

LIBRARY OF THE
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

510.84

Il 63c

no. 11-20



ENGINEERING

AUG 5 1976

The person charging this material is responsible for its return to the library from which it was withdrawn on or before the **Latest Date** stamped below.

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

ENGINEERING

CONFERENCE ROOM

AUG 16 1977

AUG. 23 1977

NOV 16 1984

510.84
I263c
no.15

Engin.

CONFERENCE ROOM

ENGINEERING LIBRARY
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS

Center for Advanced Computation

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

URBANA ILLINOIS 61801

CAC Document No. 15

ECONOMIC RESEARCH GROUP WORKING
PAPER NO. 5

The CAC Economic and Manpower
Forecasting Model:
Documentation and User's Guide

R. H. Bezdek, R. M. Lefler,
A. L. Meyers, J. H. Spoonamore



Digitized by the Internet Archive
in 2012 with funding from
University of Illinois Urbana-Champaign

<http://archive.org/details/caceconomicmanpo00bezd>

BIBLIOGRAPHIC DATA SHEET		1. Report No. UIUC-CAC-DN-71-15	2.	3. Recipient's Accession No.
4. Title and Subtitle THE CAC ECONOMIC AND MANPOWER FORECASTING MODEL: DOCUMENTATION AND USER'S GUIDE				5. Report Date October 15, 1971
7. Author(s) R.H. Bezdek, R.M. Lefler, A.L. Meyers, J.H. Spoonamore				6.
9. Performing Organization Name and Address Center for Advanced Computation University of Illinois at Urbana-Champaign Urbana, Illinois 61801				8. Performing Organization Rept. No. CAC-15
12. Sponsoring Organization Name and Address U.S. Army Research Office-Durham Duke Station Durham, North Carolina				10. Project Task Work Unit No.
				11. Contract Grant No. DAHCO4 72-C-0001
				13. Type of Report & Period Covered Research
15. Supplementary Notes				14.
16. Abstracts This paper presents the preliminary documentation and user's guide for the Center for Advanced Computation economic and manpower forecasting model. Section I gives introductory and background information on the development of the model and presents a brief but rigorous theoretical basis for the on-line system. Section II gives a description of the basic MANPOWER/DEMAND program indicating the function of the program, the detailed workings of the system options, and the language in which it is written. Appendices contain specifications of the data tapes and disc files involved, flow charts of the computer processes, and sample data input and output.				
17. Key Words and Document Analysis. 17a. Descriptors Applications Social and Behavioral Sciences Economics Forecasting (Manpower)				
17b. Identifiers/Open-Ended Terms				
17c. COSATI Field/Group				
18. Availability Statement No restriction on distribution. Available from National Technical Information Service, Springfield, Virginia 22151			19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 63
			20. Security Class (This Page) UNCLASSIFIED	22. Price

CAC Document No. 15

ECONOMIC RESEARCH GROUP WORKING PAPER NO. 5

THE CAC ECONOMIC AND MANPOWER FORECASTING MODEL:
DOCUMENTATION AND USER'S GUIDE

By

Roger H. Bezdek

R. Michael Lefler

Albert L. Meyers

Janet H. Spoonamore

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

October 15, 1971

This work was supported in part by the Advanced Research Projects
Agency of the Department of Defense and was monitored by the U.S.
Army Research Office-Durham under Contract No. DAHCO4 72-C-0001.

Approved for public release; distribution unlimited.

THE CAC ECONOMIC AND MANPOWER FORECASTING
MODEL: DOCUMENTATION AND USER'S GUIDE

by

Roger H. Bezdek

R. Michael Lefler

Albert L. Meyers

Janet H. Spoonamore

ABSTRACT

This paper presents the preliminary documentation and user's guide for the Center for Advanced Computation economic and manpower forecasting model. Section I gives introductory and background information on the development of the model and presents a brief but rigorous theoretical basis for the on-line system. Section II gives a description of the basic MANPOWER/DEMAND program indicating the function of the program, the detailed workings of the system options, and the language in which it is written. Appendices contain specifications of the data tapes and disc files involved, flow charts of the computer processes, and sample data input and output.

TABLE OF CONTENTS

I.	DESCRIPTION OF THE CAC ECONOMIC AND MANPOWER FORECASTING MODEL	1
	A. The Development of the Model	1
	B. Theoretical Basis for the Program	3
II.	DESCRIPTION OF THE MANPOWER/DEMAND PROGRAM	11
	A. Function of the Program	11
	B. Description of EDITFILE	12
	C. MANPOWER/DEMAND Processing	13
	D. The Data	17
	E. Use of MANPOWER/DEMAND	18
III.	APPENDICES	
	Appendix A: Tape Specifications	24
	Appendix B: Flow Charts	25
	Appendix C: Source List of MANPOWER/DEMAND	36
	Appendix D: Sample Input and Output	52

I. DESCRIPTION OF THE CAC ECONOMIC AND MANPOWER FORECASTING MODEL

A. The Development of the Model

The Center for Advanced Computation (CAC) economic and manpower forecasting model was initially conceived in the summer of 1969 by Roger Bezdek and Hugh Folk. At that time it was clear that there was, and would continue to be, a pressing need for a general, consistent economic model capable of analyzing both direct and indirect effects of specified changes in the economic environment on the economy and labor market. No model available was capable of simulating in detail the overall effects of changes in expenditures on different types of economic programs and activities which corresponded to alternate national priorities. The development of such a model was undertaken by Roger Bezdek for his Ph.D. thesis in Economics at the University of Illinois at Urbana-Champaign.

The Manpower Administration of the U. S. Department of Labor supported the major portion of Bezdek's dissertation research through a Doctoral Dissertation Grant. Although Bezdek originally planned to develop both historical and projected versions of this model, the latter development was prevented by severe methodological and statistical difficulties. Bezdek's original model pertained to the year 1960. Its development and the results of simulations conducted with it are described in detail in Manpower Implications of Alternate Patterns of Demand for Goods and Services.¹

¹ Bezdek [2]

In the spring of 1971, Bezdek and James Scoville developed for the National Urban Coalition a projected version of the basic model which pertained to the mid-1970's. It was used to simulate the effects on the U. S. labor market which would likely be generated by the Urban Coalition's proposed reorderings of national goals and priorities contained in Counter-budget. The model is discussed in detail in Bezdek and Scoville's Manpower Implications of Reordering National Priorities.²

Early in 1971, personnel from the newly established Center for Advanced Computation of the University of Illinois became interested in continuing work on this model at the Center. The Center for Advanced Computation is an outgrowth of the ILLIAC IV project. It is an independent unit of the Graduate College which provides an interdisciplinary environment for research projects requiring specialized and sophisticated computer facilities. The development of this type of economic model required sophisticated and efficient computer software and, from the Center's point of view, this model offered a feasible and potentially significant application for ILLIAC IV.

Agreement was reached and an Economic Research Group (ERG) was established at the Center. Bezdek spent the summer of 1971 transferring the basic model onto the Center's computer facilities and improving and expanding it. Work continues in this direction, with Bezdek supervising development of the demand side of the model and Hugh Folk directing development of the supply side. This booklet is written as a user's guide to the demand-generating portion of the CAC model.

Input components of the model include data tapes, disc files, and computer card decks which can be integrated into a number of consistent

² Bezdek and Scoville [6].

$(I-A)^{-1}$ is the Leontief inverse matrix and its elements a_{ij} indicate the output requirements generated directly and indirectly from industry i by industry j per delivery of a dollar's worth of output to final demand.

The final demand vector itself may be viewed as the sum of a number of vectors each of which represents the industrial requirements of a distinct component of final demand. Letting u denote the number of final demand activities, g_j denote an n -by-1 vector specifying the direct output requirements of exogenous activity j , and ϵ_j denote a vector indicating the portion of final demand consumed by exogenous activity j , we have:

$$(6) \quad y = g_1 + g_2 + \dots + g_u; \sum_i y_i = \sum_j \epsilon_j \left(\sum_i y_i \right); \sum_j \epsilon_j = 1$$

Writing out the first part of (6) specifically yields linear equations of the following form:

$$(7) \quad y_i = g_{i1} + g_{i2} + \dots + g_{ij} + \dots + g_{iu}; \quad i = 1, 2, \dots, n$$

Consider an arbitrary element g_{ij} defined above. As indicated, g_{ij} shows the direct requirements for input i generated by exogenous activity j and the magnitude of this demand will generally be determined by two factors: the total amount of final demand absorbed by activity j , and the portion of this amount devoted to the purchase of input i . The first factor may be expressed as: $\epsilon_j \sum_i y_i$, while the second factor is written as:

$g_{ij} / \sum_i g_{ij}$. Letting $q_j = \epsilon_j \sum_i y_i$, and $p_{ij} = g_{ij} / \sum_i g_{ij}$, equation (7) can be rewritten as:

$$(8a) \quad y_i = p_{i1}q_1 + p_{i2}q_2 + \dots + p_{ij}q_j + \dots + p_{iu}q_u; \quad i = 1, 2, \dots, n$$

or, letting P denote an n -by- u activity-industry matrix of activity input coefficients, and letting q denote a u -by-1 activity-expenditure vector:

$$(8b) \quad y = Pq$$

p_{ij} indicates the direct requirements generated for the output of industry i per dollar of expenditure in final demand sector j , and q_j shows the amount of expenditures allocated to activity j .

Within this framework it is possible to determine the direct output requirements generated by alternate distributions of national expenditures among economic activities. Here it is assumed that the elements of the P matrix are fixed over a limited range of expenditure redistribution; the activity-industry matrix thus represents a transformation of expenditures on economic activities into direct output requirements from every industry in the economy. Using equation (5), these direct output requirements can be translated into total output requirements from every industry.

Next, output requirements must be related to employment demands. To accomplish this it is assumed that the employment requirements of an industry are proportional to the industry's output and that this relationship may be expressed in terms of labor input coefficients. Letting x_i^e denote the total employment in industry i , the labor input coefficient for industry i , θ_i , is:

$$(9) \quad \theta_i = x_i^e / X_i; \quad i = 1, 2, \dots, n$$

Labor input coefficients are thus derived by dividing industry employment by industry output and they show the employment requirements of an industry per unit of output. Employment in each industry may be related to the components of final demand by substituting the values given for X_i in (5) into equation (9). Equations of the following form are derived:

$$(10a) \quad x_i^e = \theta_i \hat{a}_{i1} y_1 + \theta_i \hat{a}_{i2} y_2 + \dots + \theta_i \hat{a}_{ij} y_j + \dots + \theta_i \hat{a}_{in} y_n;$$

$$i = 1, 2, \dots, n$$

or, letting x^e denote an n-by-1 vector of elements $x_1^e, x_2^e, \dots, x_n^e$, and letting Θ denote a diagonal matrix whose elements are $\theta_1, \theta_2, \dots, \theta_n$, the equations in (10a) may be written in matrix notation as:

$$(10b) \quad x^e = \Theta (I-A)^{-1} y$$

Consider the matrix M defined as $M = \Theta(I-A)^{-1}$ whose elements m_{ij} are:

$$(11) \quad m_{ij} = \theta_i \hat{a}_{ij}; \quad i, j = 1, 2, \dots, n$$

Any element m_{ij} of M shows the total employment required within industry i in order for industry j to deliver a dollar's worth of output to final demand. Each row of M indicates the manner in which employment is generated within industry i by required activity in industries 1, 2,, n and each column of M illustrates how the employment generated by industry j is distributed among all industries. This matrix is referred to as an inter-industry-employment matrix.

The necessary theoretical framework has now been constructed which permits the transformation of alternate priority-expenditure distributions into distinct interindustry-employment demand patterns. Letting $\bar{Y}^*$ denote an n-by-n diagonal final demand matrix, the "total" interindustry-employment matrix, M^T , is derived by postmultiplying M by $\bar{Y}^*$:

$$(13) \quad M^T = M \bar{Y}^*$$

The elements of M^T show the total employment generated by and within every industry for a generated distribution of final demand reflecting a specified priority alternative.

The final step in the construction of the theoretical model involves the relation of interindustry-employment requirements to demands for occupational categories of manpower resources. This transformation is accomplished by using an industry-occupation matrix showing the occupational

distribution of industry employment for the time period under consideration. Denote this matrix by B: the rows of B represent industries, the columns of B represent occupations, and any element b_{ik} of B shows the percent of total employment in industry i composed of persons classified within occupation k.

Let R denote a diagonal matrix whose elements r_{ii} are the row sums of the interindustry-employment matrix and thus show the total employment generated within industry i. One type of manpower information is derived by premultiplying the industry-occupation matrix by R:

$$(14a) \begin{bmatrix} r_{11} & & & & \\ & r_{22} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & r_{nn} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & \dots & \dots & b_{1h} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{n1} & b_{n2} & \dots & \dots & b_{nh} \end{bmatrix} = \begin{bmatrix} s_{11}^{(\alpha)} & s_{12}^{(\alpha)} & \dots & \dots & s_{1h}^{(\alpha)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{n1}^{(\alpha)} & s_{n2}^{(\alpha)} & \dots & \dots & s_{nh}^{(\alpha)} \end{bmatrix}$$

or

$$(14b) \quad RB = S^{(\alpha)}$$

$S^{(\alpha)}$ is a "type α " interindustry-occupation matrix and the elements $s_{ik}^{(\alpha)}$ of it show the total demands for occupation k generated within industry i by a specified distribution of national expenditures.

Letting $\bar{M}^0$ denote the transpose of the total interindustry-employment matrix, a second type of manpower impact matrix is derived by premultiplying the industry-occupation matrix by $\bar{M}^0$:

(15a)

$$\begin{bmatrix} m_{11} & m_{21} & \cdot & \cdot & \cdot & \cdot & m_{n1} \\ \cdot & & & & & & \cdot \\ \cdot & & & & & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & & & & \cdot \\ m_{1n} & m_{2n} & \cdot & \cdot & \cdot & \cdot & m_{nn} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & \cdot & \cdot & \cdot & \cdot & b_{1h} \\ \cdot & & & & & & \cdot \\ \cdot & & & & & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & & & & \cdot \\ b_{n1} & b_{n2} & \cdot & \cdot & \cdot & \cdot & b_{nh} \end{bmatrix} = \begin{bmatrix} s_{11}^{(\beta)} & s_{12}^{(\beta)} & \cdot & \cdot & \cdot & \cdot & s_{1h}^{(\beta)} \\ \cdot & & & & & & \cdot \\ \cdot & & & & & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & & & & \cdot \\ s_{n1}^{(\beta)} & s_{n2}^{(\beta)} & \cdot & \cdot & \cdot & \cdot & s_{nh}^{(\beta)} \end{bmatrix}$$

or:

$$(15b) \quad \bar{M} = S^{(\beta)}$$

$S^{(\beta)}$ is referred to as a "type β " interindustry-occupation matrix and the elements $s_{ik}^{(\beta)}$ of it show the demands for occupation k generated by industry i . So while the type α manpower matrix indicates the occupational employment demand generated in every industry, the type β manpower matrix indicates the occupational employment demands generated by every industry.

Finally, a third type of manpower impact matrix can also be derived. Letting B^k denote an n -by- n diagonal matrix whose elements correspond to the k^{th} column of B , the third type of manpower matrix is derived by premultiplying B^k by the transposed total interindustry employment matrix:

(16a)

$$\begin{bmatrix} m_{11} & m_{21} & \cdot & \cdot & \cdot & m_{n1} \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & & & \cdot \\ m_{1n} & m_{2n} & \cdot & \cdot & \cdot & m_{nn} \end{bmatrix} \cdot \begin{bmatrix} b_{11}^{(k)} & & & & & \\ & b_{22}^{(k)} & & & & \\ & & \bigcirc & & & \\ & & & \ddots & & \\ \bigcirc & & & & & b_{nn}^{(k)} \end{bmatrix} = \begin{bmatrix} s_{11}^{(k)} & s_{12}^{(k)} & \cdot & \cdot & \cdot & s_{1n}^{(k)} \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & & & \cdot \\ s_{n1}^{(k)} & s_{n2}^{(k)} & \cdot & \cdot & \cdot & s_{nn}^{(k)} \end{bmatrix} \quad k = 1, 2, \dots, h$$

or:

$$(16b) \quad o_{MB}^{(k)} = S^{(k)} ; k = 1, 2, \dots, h$$

Since there are h columns in B--one for each occupational classification--it is possible to derive h of these S^k matrices. Each S^k matrix is essentially an interindustry-employment matrix for the k^{th} occupation, and an element s_{ji}^k shows the employment requirements for occupation k generated within industry i by industry j. These matrices are referred to as occupational employment profiles, and they contain a highly detailed description of the structure of demands generated for an individual occupation by a specified distribution of national expenditures.

Taken together, these three types of manpower impact matrices provide a comprehensive and highly detailed picture of the employment impacts likely to result from the implementation of alternate types of economic and social programs and priorities.

II. DESCRIPTION OF THE MANPOWER/DEMAND PROGRAM⁴

A. Function of the Program

The MANPOWER/DEMAND program performs two data handling functions. First, it edits existing data structures, the input matrices to the model. Secondly, and most importantly, it performs the algebraic computations set forth by the theoretical model previously described, for generation of manpower demands based on alternative expenditure patterns and technological assumptions. In effect, the program permits the researcher to experiment by varying the data matrices which represent the input to the economic model and to study the results generated by the MANPOWER/DEMAND processes. For example:

The experimenter executes the general model for a given set of 58 proposed expenditure alternatives, noting the generated occupational employment. By changing the activity-expenditure elements to represent a different pattern of resource allocation, he can analyze the generated effects on the labor market produced by the program. In this case, he must modify the q-vector which represents the expenditure distribution. After observing these results, he may then modify the interindustry-employment matrix or the industry-occupation matrix. Another run on the model gives different results and insight into more modifications.

The MANPOWER/DEMAND program is essentially a model of economic processes represented by several matrix operations but, in addition, its flexibility permits modification of the input matrices prior to execution of these operations.

⁴ The CAC model described in this report is in the process of being expanded and improved. At periodic intervals additional documentation and user guide reports shall be published which specify the changes which have been made.

B. Description of EDITFILE

Editing is performed on files prior to execution of the MANPOWER/ DEMAND routine. The following modifications can be accomplished for each matrix:

1. Element-by-element addition, subtraction, multiplication, or division.
2. Overwriting a column or columns with a new column or columns.
3. Deletion of a column or columns, thereby reducing the size of the matrix.
4. Insertion of a new column or columns between existing ones, thereby increasing the size of the matrix.
5. Scaling an entire matrix by multiplying each row by a given constant.

EDITFILE is invoked by the standard ALGOL subroutine call as follows:

```
EDITFILE(<File name>,<number of cols>,<number of rows>);
```

It accepts on card input the following commands in free form:

```
ADD(<column number>)
SUBTRACT (<column number>)
MULTIPLY(<column number>)
DIVIDE(<column number>)
DELETE(<column number>)
INSERT(<column number>)
SCALE
```

After each command, data cards are included which contain the operands for the above commands in free-field format, i.e., integers or decimal numbers separated by commas. This routine is not invoked if editing of existing matrices is not desired.

C. MANPOWER/DEMAND Processing

Program Overview

ERGWORKS is the routine which performs the major operations dictated by the system. It can accept as input different vectors reflecting various levels and distributions of expenditures on economic activities and it returns generated employment requirements classified by industry or by occupation. Alternately, the expenditure vector can be held constant and manpower demands can be generated by changing rows, columns, or individual coefficients within the various matrices to reflect changes in technology, labor productivity, or occupational displacement. ERGWORKS presently consists of four distinct sections: a core section, which is always executed, and three branches, only one of which is executed during a run. The choice of branch is a user-input control option and depends upon what information the user wants the program to calculate. Branch three, for example, selects a single occupation and gives detailed information on the structure of demand for that occupation.

The Core Section

The first step is to read in and check the list of control options provided by the user. The first option is the year. If the user asks for 1972, the program calls in the data from the four disk files corresponding to 1972's data; if the user asks for 1976, the program calls in the 1976 projected data. If the user asks for a year other than 1972 or 1976, the program will form the three required matrices and the required q-vector by performing a standard linear interpolation of the data for both years. The interpolation is performed by a special subroutine which subtracts 1972 from the input year, then multiplies the difference between the 1976 data and

the 1972 data by one fourth of the difference in the years and adds the result to the 1972 data. It should be noted, however, that linear interpolation may not necessarily represent economic change within an interindustry model.

The next option read in indicates whether the user wants only the final results and certain selected intermediate results or whether he also wants all intermediate matrices printed out in full. The third option specifies which branch the program is to take. If this option is three, the program also reads in a column number corresponding to the column of that occupational classification in the B-matrix. The program tests both the branch option and the column to assure that the former lies within the range one to three and that the latter lies within the range one to one hundred eighty-five.

The program then reads in the user-given title of the run and the q -vector. Both are printed immediately. Next, each of the fifty-eight rows of P^T is multiplied by the corresponding element of q . This is mathematically equivalent to converting q into a diagonal matrix and post-multiplying this diagonal matrix by P^T . If the "fullprint" option has been called by the user, this new 58×89 matrix is printed, first by columns, then by rows.

The same basic code is used to print out each of the matrices required by the fullprint option. The title of the matrix is written, and a FOR-loop selects columns in groups of ten until less than ten columns remain. For each of the rows of the matrix the ten elements corresponding to the ten columns are printed. When less than ten columns remain to be written, the routine prints the end of each row of the matrix, the number of elements printed corresponding to the number of columns left. Once this is done, the same operation is carried out for the transpose of the matrix, effectively causing the matrix to be written by rows instead of columns.

After the new 58 x 89 matrix is printed (if it is to be printed), a y-vector is created which is eighty-nine elements long. The elements of the y-vector are the column sums of the 58 x 89 matrix.

The y-vector is printed and aggregated to eighty-five elements by deleting four selected elements, and the aggregated vector is printed.

Each of the columns of M is then multiplied by the corresponding element of the y-vector. This corresponds mathematically to post-multiplying the M-matrix by a diagonal matrix created from the y-vector. The row sums and column sums of this new matrix are computed and printed and, if the fullprint option is on, the entire matrix is printed.

After the above operation is completed the new M-matrix is aggregated to a 66 x 85 matrix. Row sums and column sums of this aggregated matrix are taken and a sixty-six order vector, designated r, is created from the sixty-six row sums. To avoid double-counting, the column sums are actually calculated over selected rows of the M-matrix. The row sums, the column sums, the sums of selected elements of both, and (if the fullprint option is on) the matrix itself are printed.

If the program is to branch to the second or third branch, the last operation completed by this core section consists of multiplying each row of the aggregated M-matrix by the corresponding element of the μ -vector and (if the fullprint option is on) printing the resultant matrix.

Branch One

Branch one requires the r-vector computed above. But each element of this vector is first multiplied by the corresponding element of the μ -vector. Each row of the B-matrix is then multiplied by the corresponding element of the modified r-vector, forming a matrix called $S^{(\alpha)}$. If the fullprint option is on $S^{(\alpha)}$ is printed. In both cases row sums and column sums are calculated

for $S^{(\alpha)}$ over selected columns and rows, respectively, to avoid double-counting. These row sums and column sums are printed and totaled and the program run then terminates.

Branch Two

Branch two postmultiplies the transpose of the modified aggregated M-matrix by the B-matrix, producing an 85×185 matrix called $S^{(\beta)}$. This operation is modified, however, by multiplying only selected columns and rows to avoid double-counting. If the fullprint option is on, the $S^{(\beta)}$ matrix is printed out.

The one hundred eighty-five column sums are computed and totaled (over selected rows to avoid double-counting), then printed. Similarly, the eighty-five row sums are computed and totaled (over selected columns to avoid double-counting), then printed. This terminates execution.

Branch Three

Branch three begins by selecting and printing a column from the B-matrix. Each of the sixty-six columns of the transpose of the modified aggregate M-matrix is then multiplied by the corresponding element of this column vector to form an 85×66 matrix called $S^{(k)}$. In the $S^{(k)}$ matrix h represents the column of the B-matrix selected, where $1 \leq k \leq 185$. If the fullprint option is on, this matrix is printed.

The sixty-six column sums are computed and totaled (over selected rows to avoid double-counting), then printed. Similarly, the eighty-five row sums are computed and totaled (over selected columns to avoid double-counting), then printed. This terminates execution.

D. The Data

The input data for both routines reside on the same set of disk and card files. The eight disk files contain projected data for years 1972 and 1976 derived from data which were obtained from the Office of Business Economics, the Bureau of Labor Statistics, the Harvard Economic Research Project, the National Planning Association, and the National Urban Coalition, and which were, in part, derived independently by Roger Bezdek. These disk files are matrix representations for the 58 x 89 activity-industry matrix, designated by "P", the 85 x 85 interindustry-employment matrix, designated by "M", the 66 x 185 industry-occupation matrix, designated by "B", and a 66-order vector designated by " μ ".

The P and the M matrices are stored and handled in transposed form within the program. These files can be inputted directly to the model or modified first by EDITFILE and then used as direct inputs. The card file called CARD contains, first of all, the specifications for any editing to be done on the above disk files. Following these specifications are the run time options for ERGWORKS. These options include the projected year to be run, the fullprint option, and the branch of the routine to be invoked. Following the option, the q-vector (the specified expenditure distribution) is read in.

E. Use of MANPOWER DEMAND

Tapes:

MANPOWER/DEMAND is written in Burroughs B6500 ALGOL language. The source program as well as all data files are stored on nine-track tapes in the B6500 room, Room 10, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois. At present, several versions of the program are saved at points in time to enable programming changes to be made without risk to previous progress. In the appendix a list provides tape numbers and tape contents.

The tape name ERG is used to access any of the tapes. In this and any discussion of B6500 usage, the reader is referred to the Little Golden Book of the B6500⁵ for operating system details.

Control Cards:

In order to run MANPOWER/DEMAND on the B6500, a set of control statements must be entered in card form or from a terminal. Listed below are the cards which can be used:

The tape must first be loaded onto disk by the following:

?COPY ERG/= FROM ERG

In order to compile the program:

?COMPILE ERG/MANPOWER/DEMAND WITH ALGOL LIBRARY

?ALGOL FILE CARD = ERG/OCTI DISK SERIAL

?END

The execution is accomplished by the following:

?EXECUTE ERG/MANPOWER/DEMAND

?DATA CARD

⁵ Abel [1].

2, 1, 1,

TEST DATA

103351 48160

and other data cards

?END

(Note: ? is a control character on B6500 and is punched by
multipunch 1, 2, 3.)

The execution takes about six minutes of processing time on the B6500 which
amounts to about 15 minutes real time in the machine.

Sample Experiments:

Statement of Problem 1: Replace the first 12 columns of the 1972
P matrix with a given set of data, change column 17 to 17a and 17b
and run the program for branch one.

Method of Solving: Use the EDITFILE procedure to modify the
ERG/MATRIX/P1 file. The following code must be inserted into the
program to form executable code.

```
L: = 57  
EDITFILE (P1, L, 89);  
REWIND (P1);  
ERGWORKS (L, 89, 85, 66, 185);  
END
```

The program is then recompiled, and executed with the following card input:

```
?DATA CARD  
REPLACE (1)  
(Data cards to replace second column)  
REPLACE (2)  
(Data cards to replace second column)  
.  
.  
.  
REPLACE (12)
```



```
(Data cards to replace 12th column)

REPLACE (17)

(Data cards to replace 17th column)

INSERT (18)

(Data cards to insert between the 17th and 18th columns--
changes old 18th to 19th column)

STOP

72,1, 1

TEST DATA

103351 418160 ...

(Change 57 to 58 long vector here)

14561 ...

?END
```

Statement of Problem 2: Scale the rows of the 1972 M matrix by a set of constants. Run the program using the modified P matrix above for branch one.

Method of Solving: Again use the EDITFILE procedure to modify the ERG/MATRIX/M3 file. If possible, store the modified P-matrix from the last example; otherwise, include the changes here as in the above program.

The program instructions follow:

```
L: = 58;

EDITFILE (M3, 85, 85);

REWIND (M3);

ERGWORKS (L, 89, 85, 66, 185);

END
```

The above statements are to be inserted in the executable code section of the program. Recompile and execute with the following:

```
?DATA CARD

MULTIPLY (1)
```

(85 elements to multiply row 1)

MULTIPLY (2)

(85 elements to multiply row 2)

.

.

.

MULTIPLY (85)

(85 elements to multiply row 85)

STOP

72, 1, 1,

TEST DATA

103351 418160 ...

.

.

.

14561 ...

?END

REFERENCES

- [1] Abel, Norma. "The Little Golden Book of the B6500." ILLIAC IV Project Report, University of Illinois at Urbana-Champaign, Urbana, Illinois, June 1971.
- [2] Bezdek, Roger H. Manpower Implications of Alternate Patterns of Demand for Goods and Services. Ph. D. Thesis and report prepared for the Manpower Administration of the U. S. Department of Labor, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1971.
- [3] _____. "Manpower Implications of Alternate Patterns of Demand For Goods and Services." 1970 Proceedings of the Business and Economics Section of the American Statistical Association, pp. 417-422.
- [4] _____. Progress Report on the Development of a Large-Scale Conditional Consistent Economic and Manpower Forecasting Model. Economic Research Group Working Paper no. 1, Center for Advanced Computation Document no. 7, University of Illinois at Urbana-Champaign, Urbana, Illinois, July 1971.
- [5] _____. Manpower Analysis Within an Interindustry Framework: Theoretical Potential and Empirical Problems. Economic Research Group Working Paper no. 4, Center for Advanced Computation Document no. 13, University of Illinois at Urbana-Champaign, Urbana, Illinois, September 1971.
- [6] _____, and Scoville, James G. Manpower Implications of Reordering National Priorities. Washington, D.C.: National Urban Coalition, 1971.

- [7] Burroughs B6500 Extended Algol Language Information Manual.

Document no. 5000128, Burroughs Corporation, 1971.

- [8] McCracken, Daniel D. A Guide to Algol Programming. New York:

John Wiley and Sons, 1962.

- [9] Meyers, Albert L. An Introduction to the Pointer Mechanism in

Burroughs Corporation Algol. ILLIAC IV Document no. 215,

University of Illinois at Urbana-Champaign, Urbana, Illinois,

May 1970.

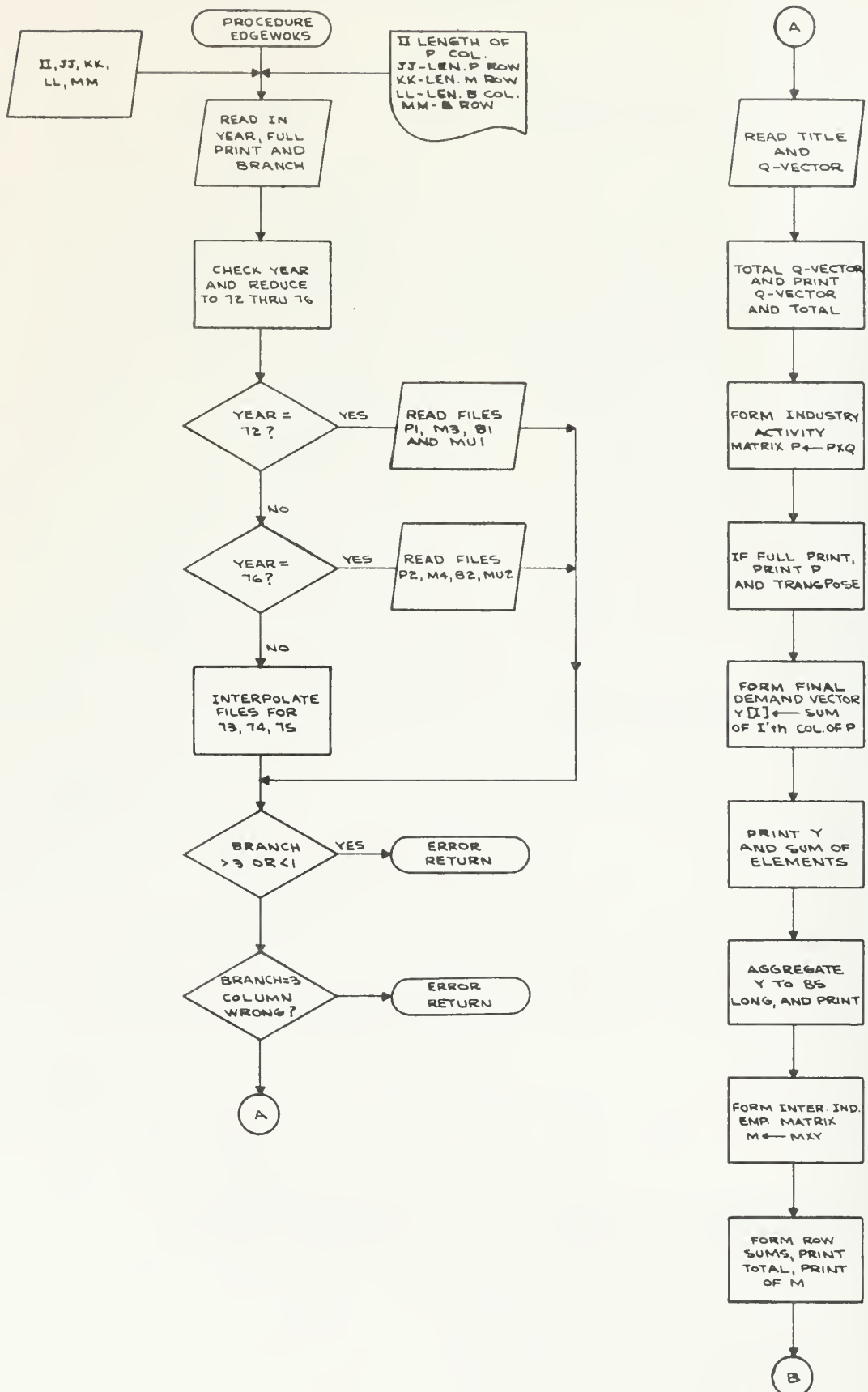
Appendix A: Tape Specifications

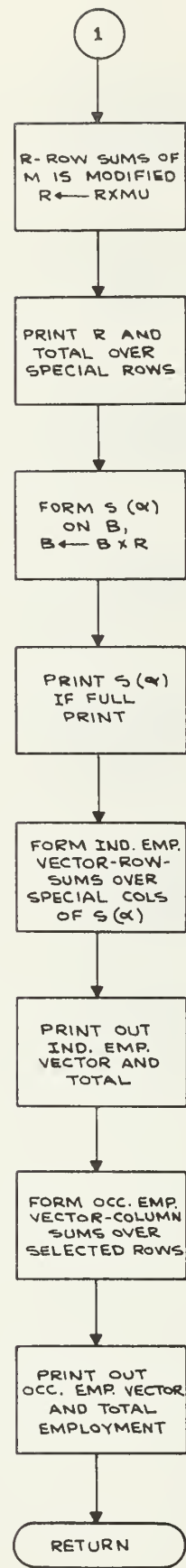
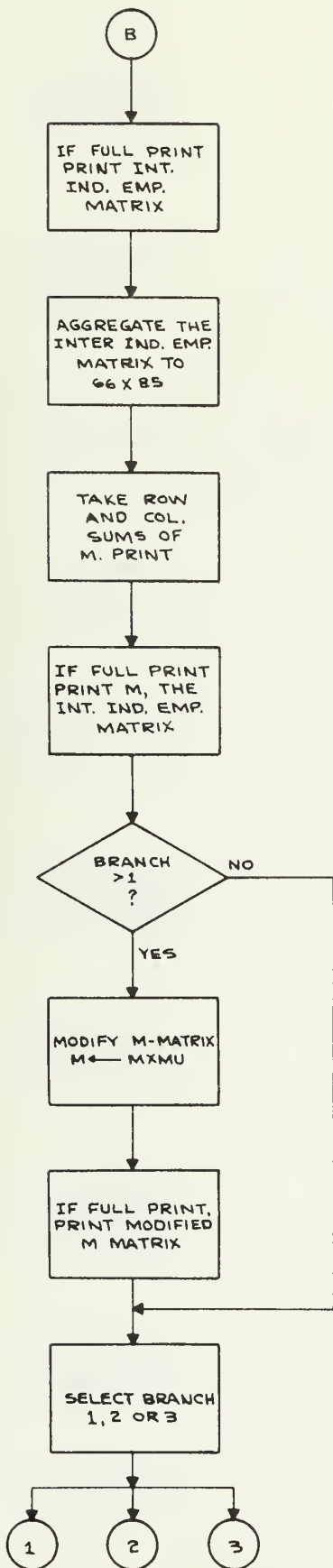
B6500

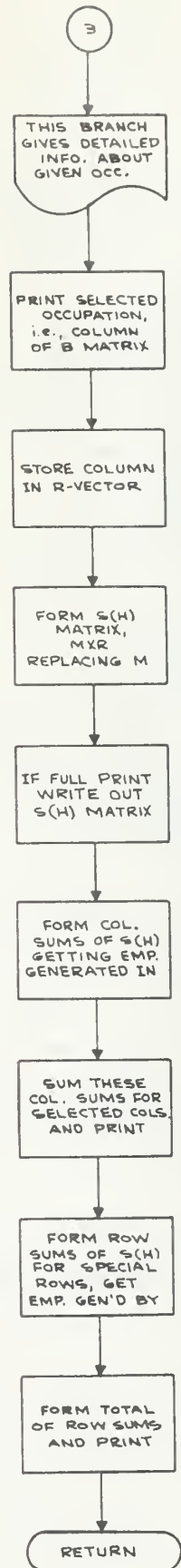
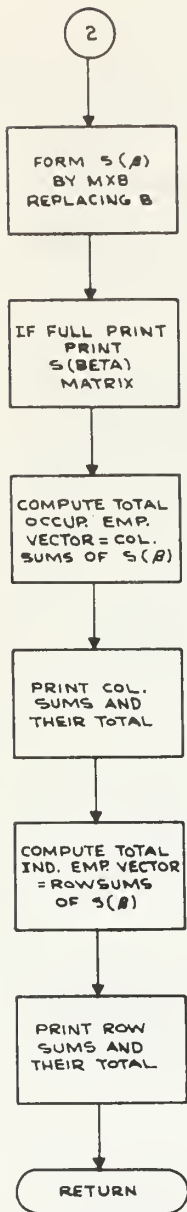
<u>Tape Number</u>	<u>Name</u>	<u>File Names</u>	<u>Description of File</u>
202,204	ERG	(1) ERG/MATRIX/P1	1972 P-MATRIX
		(2) ERG/MATRIX/P2	1976 P-MATRIX
		(3) ERG/MATRIX/M3	1972 M-MATRIX
		(4) ERG/MATRIX/M4	1976 M-MATRIX
		(5) ERG/MATRIX/B1	1972 B-MATRIX
		(6) ERG/MATRIX/B2	1976 B-MATRIX
		(7) ERG/MATRIX/MU1	1972 - Mu vector
		(8) ERG/MATRIX/MU2	1976 - Mu vector
		(9) ERG/MANPOWER/SOURCE	Sourcecode for economic model
		(10) ERG/MANPOWER/DEMAND	Compiled version of ERG/MANPOWER/DEMAND
122	ERG	files 1.-9 same as 202, 204	
		(10) ERG/OCT1	Source code for economic model and edit routine.
		(11) ERG/MANPOWER/DEMAND	Compiled version of (10.)
568	ERG	files 2.-11 same as 122	
		(1) ERG/MATRIX/P1	modified in experiment

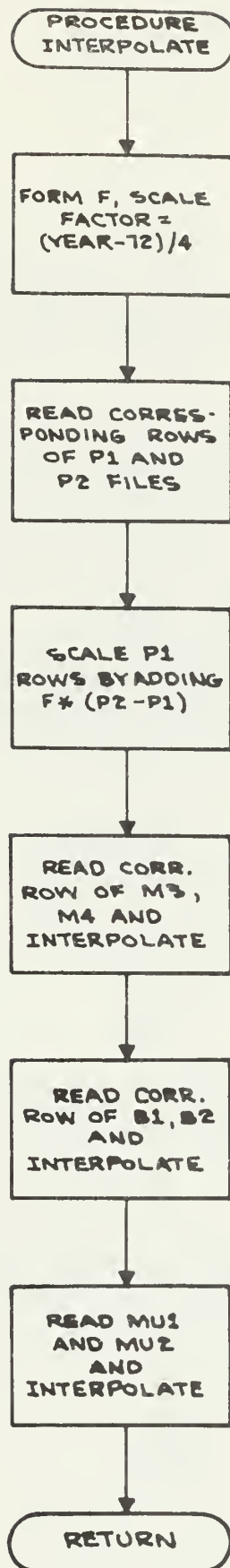
Appendix B

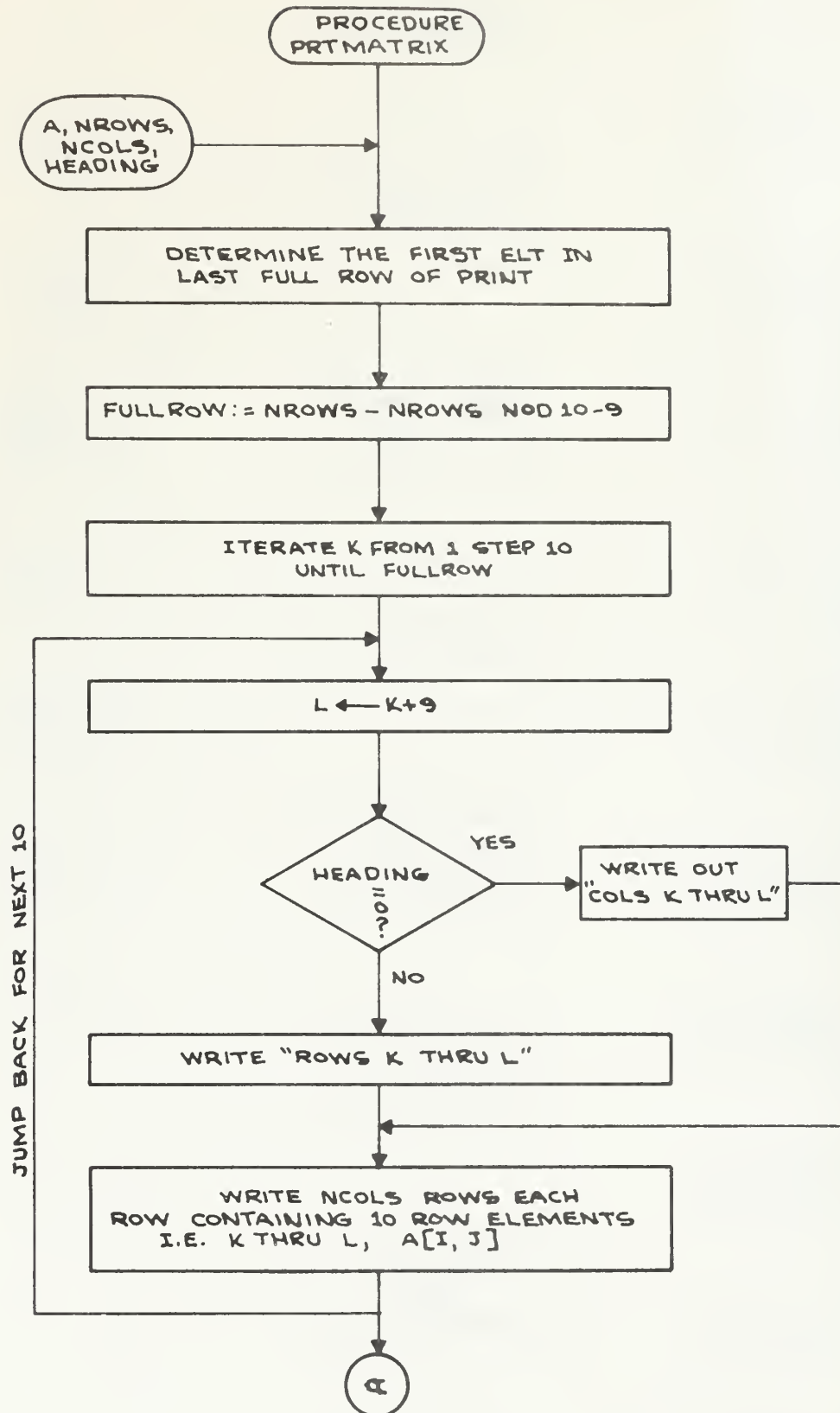
Flow Charts

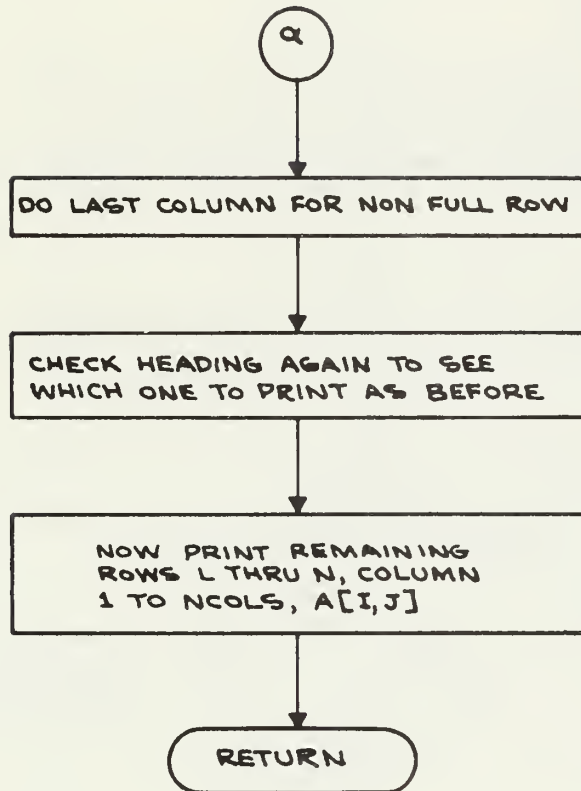


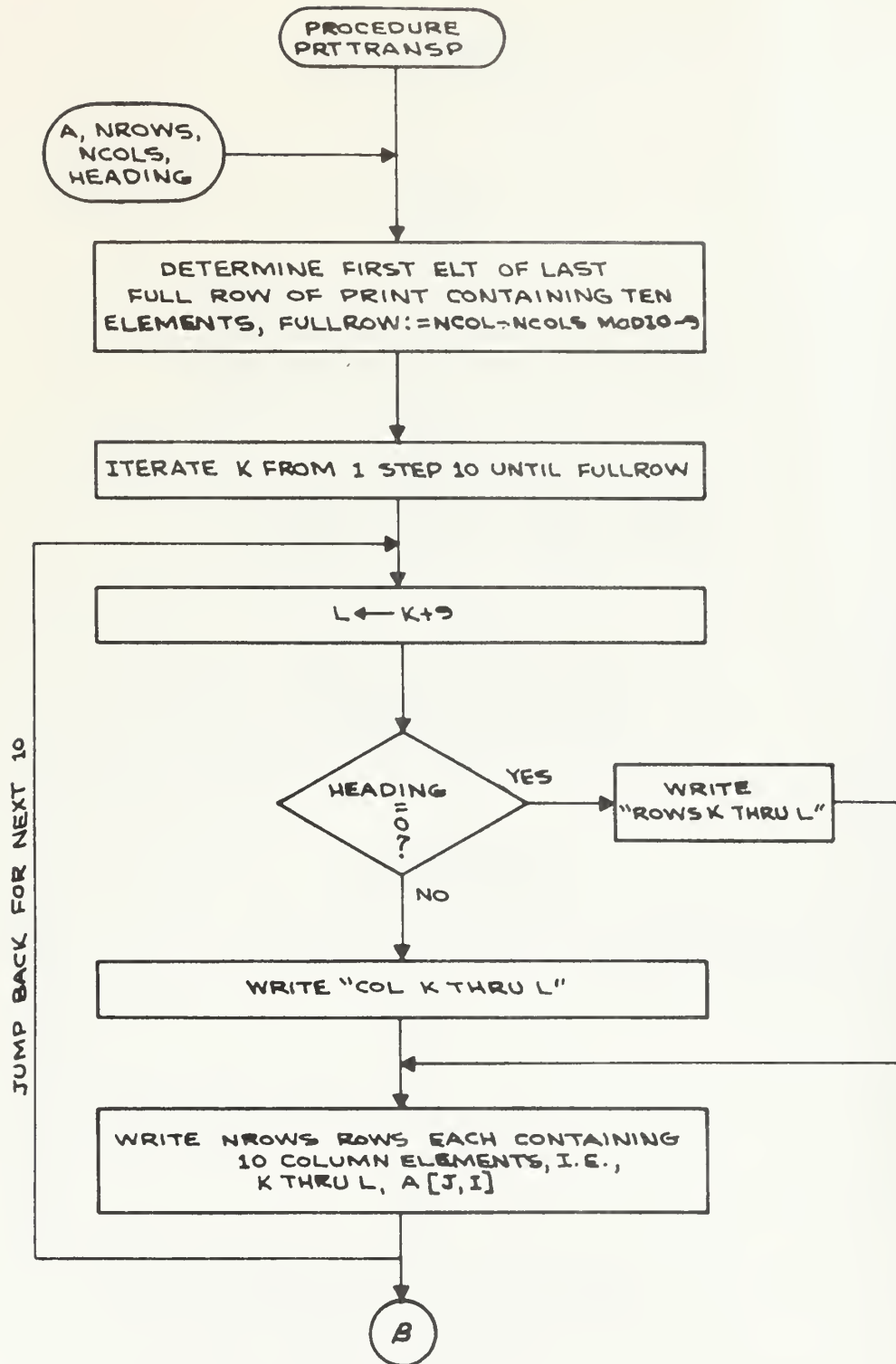


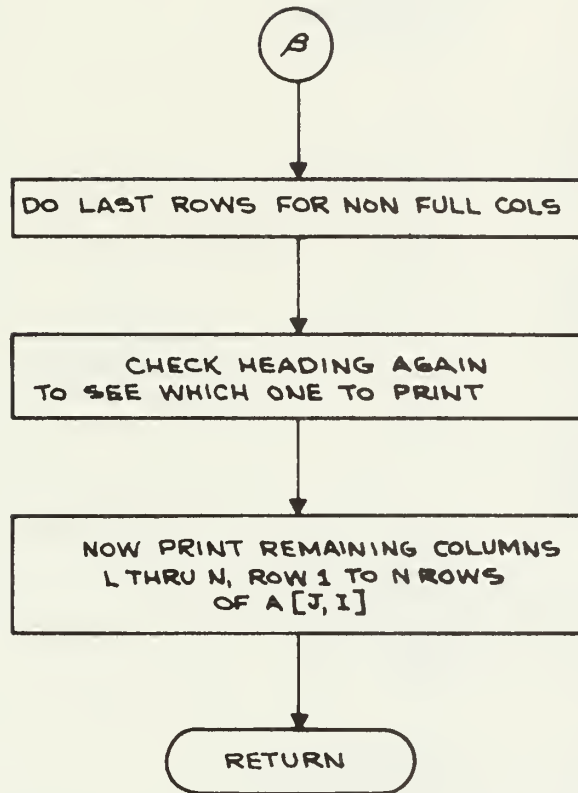


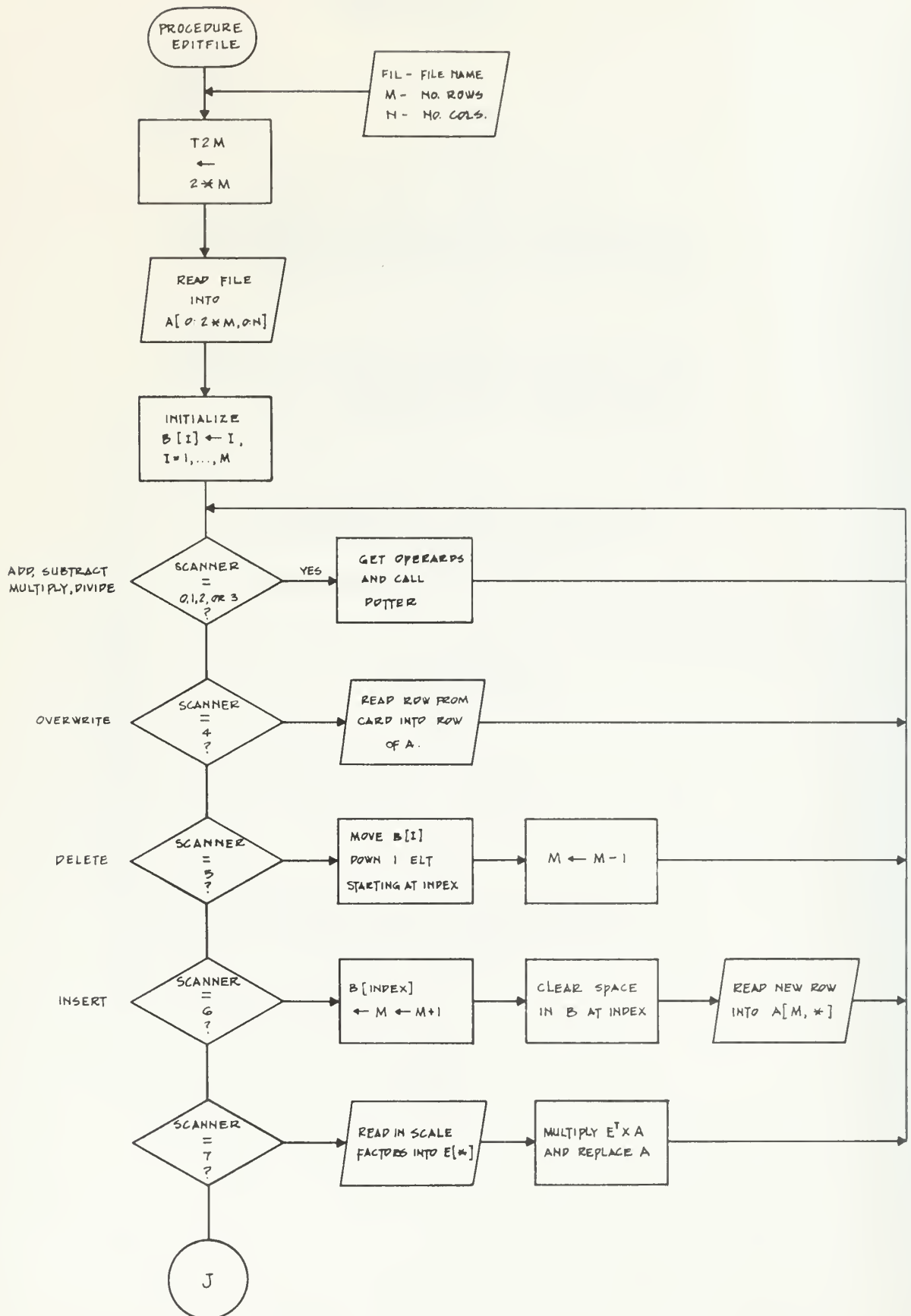


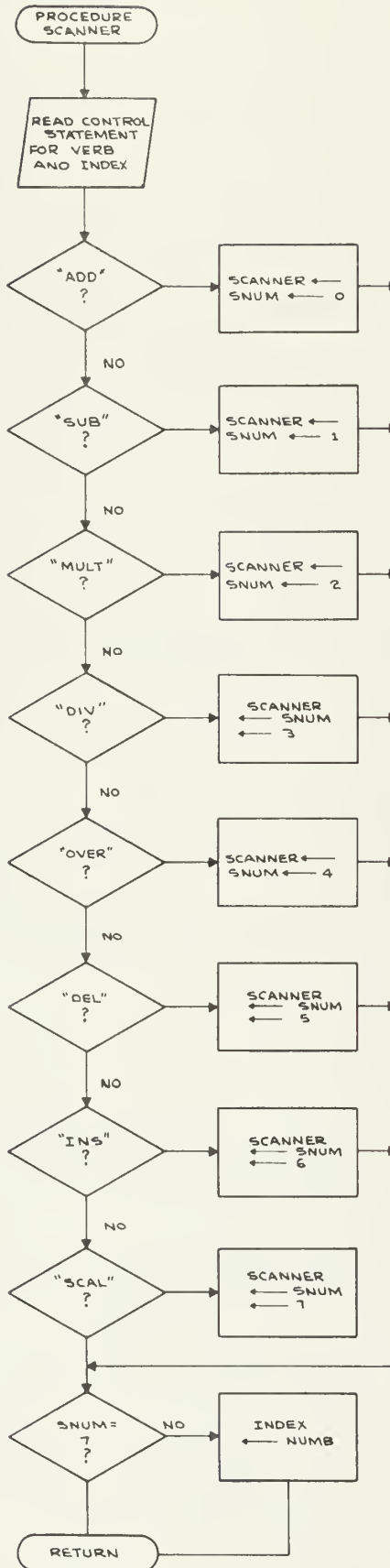
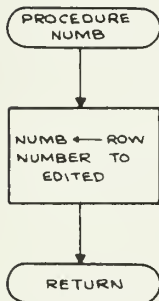
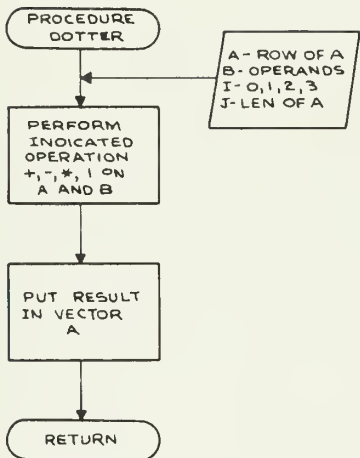












Appendix C

Source List of MANPOWER/DEMAND

```

BEGIN
DEFINE SEG = LABEL NEWSEGMENT;
FILE LINE(KIND = 135,BUFFERS = 2,MAXRECSIZE = 22,INTMODE = 4,MYUSE = 2);
FILE CARD(KIND = 9,BUFFERS = 2,MAXRECSIZE = 14,INTMODE = 4);
FILE P1(KIND = 1,ACCESS = 0,TITLE = "ERG/MATRIX/P1.",BUFFERS = 2,
MAXRECSIZE = 14,BLOCKSIZE = 420);
FILE P2(KIND = 1,ACCESS = 0,TITLE = "ERG/MATRIX/P2.",BUFFERS = 2,
MAXRECSIZE = 14,BLOCKSIZE = 420);
FILE M3(KIND = 1,ACCESS = 0,TITLE = "ERG/MATRIX/M3.",BUFFERS = 2,
MAXRECSIZE = 14,BLOCKSIZE = 420);
FILE M4(KIND = 1,ACCESS = 0,TITLE = "ERG/MATRIX/M4.",BUFFERS = 2,
MAXRECSIZE = 14,BLOCKSIZE = 420);
FILE B1(KIND = 1,ACCESS = 0,TITLE = "ERG/MATRIX/B1.",BUFFERS = 2,
MAXRECSIZE = 14,BLOCKSIZE = 420);
FILE B2(KIND = 1,ACCESS = 0,TITLE = "ERG/MATRIX/B2.",BUFFERS = 2,
MAXRECSIZE = 14,BLOCKSIZE = 420);
FILE MU1(KIND = 1,ACCESS = 0,TITLE = "ERG/VECTOR/MU1.",
BUFFERS = 2,MAXRECSIZE = 14,BLOCKSIZE = 420);
FILE MU2(KIND = 1,ACCESS = 0,TITLE = "ERG/VECTOR/MU2.",
BUFFERS = 2,MAXRECSIZE = 14,BLOCKSIZE = 420);
FORMAT IN F8(8I10),F9(X5,9F8.4,X3);
FORMAT OUT HEAD1("/ELEMENT      VALUE"), F1(I4,F20.5),
HEAD2("/      ROW",X7,"ROWSUM"), F2("TOTAL",F19.5),
HEAD3("/COLUMN      COLUMNSUM"), F7(I3,9(X1,R11.4)),
HEAD4("/COLS",I4," THRU",I4/),
HEAD5("/ROWS",I4," THRU",I4/), F10(I3,10(X1,R11.4)),
FRRMESS1("***INCORRECT OPTION OR BRANCH");
REAL ARRAY P(0:58,0:89),M(0:85,0:35);
INTEGER I,J,K,L;
LABEL F0F0;
*
* LINE PRINT ROUTINES
*
* PROCEDURES PRMATRIX AND PRTRANSP ARE USED FOR FULLPRINT
* TO PRINT MATRIX OR ITS TRANSPOSE
PROCEDURE PRTRANSP(A,NROWS,NCOLS,HEADING);
VALUE NROWS,NCOLS,HEADING;
REAL ARRAY A(0,0);
INTEGER NROWS,NCOLS,HEADING;
BEGIN
INTEGER FULLROW,I,J,K,L;
FULLROW := NCOLS * NCOLS MOD 10 = 9;
FOR K:= 1 STEP 10 UNTIL FULLROW DO BEGIN
L:= K + 9;
IF HEADING=0 THEN WRITE(LINE,HEAD5,K,L) ELSE
WRITE(LINE,HEAD4,K,L);
FOR J:= 1 STEP 1 UNTIL NROWS DO
WRITE(LINE,F10,J,FOR I:= K STEP 1 UNTIL L
DO A(J,I));
END;
IF NCOLS > L THEN BEGIN
IF HEADING = 0 THEN WRITE(LINE,HEAD5,L+1,NCOLS) ELSE
WRITE(LINE,HEAD4,L+1,NCOLS);
FOR J := 1 STEP 1 UNTIL NROWS DO
WRITE(LINE,F7,J,FOR I:= L+1 STEP 1 UNTIL NCOLS DO A(J,I));
END;
WRITE(LINE(SKIP 1));

```

```

END PRTRANSP;
PROCEDURE PRMATRIX(A,NROWS,NCOLS,HEADING);
VALUE NROWS,NCOLS,HEADING;
INTEGER NROWS,NCOLS,HEADING;
REAL ARRAY A[0,0];
BEGIN
  INTEGER FULLROW,I,J,L,K;
  FULLROW:= NROWS - NROWS MOD 10 - 9;
  FOR K:= 1 STEP 10 UNTIL FULLROW DO BEGIN
    L:= K + 9;
    IF HEADING = 0 THEN WRITE(LINE,HEAD4,K,L) ELSE
      WRITE(LINE,HEAD5,K,L);
    FOR J:= 1 STEP 1 UNTIL NCOLS DO
      WRITE(LINE,F10,J,FOR I:= K STEP 1 UNTIL L DO A[I,J]);
    END;
    IF NROWS> L THEN BEGIN
      IF HEADING = 0 THEN WRITE(LINE,HEAD4,L+1,NROWS) ELSE
        WRITE(LINE,HEAD5,L+1,NROWS);
      FOR J:= 1 STEP 1 UNTIL NCOLS DO
        WRITE(LINE,F7,J,FOR I:= L+1 STEP 1 UNTIL NROWS DO A[I,J]);
      END;
      WRITE(LINE[SKIP 1]);
    END PRMATRIX;
  
```

%
%
%

END OF LINE PRINT ROUTINES

```

PROCEDURE EDITFILE(FIL,M,N);
VALUE          N ;
INTEGER        M,N ;
FILE           FIL ;
BEGIN
  %
  LABEL OUT ,EOFA
  ;
  %
  ARRAY A[0;2*M,0;N]
    ,B[0;2*M]
    ,C[0;N]
    ,D[0;13]
    ,E[0;2*M]
  ;
  %
  POINTER P
    ,Q
  ;
  %
  INTEGER I
    ,J
    ,INDEX
    ,SNUM
    ,T2M
  ;
  %
  DEFINE
    G(I,J) = A[B[I],J]*
    ,GETC   = READ(CARD,/,FOR I:=1 STEP 1 UNTIL N DO C[I])#
    ,DOTS   = BEGIN
  
```

%

LABEL OUT ,EOFA

;

%

ARRAY A[0;2*M,0;N]

,B[0;2*M]

,C[0;N]

,D[0;13]

,E[0;2*M]

;

%

POINTER P

,Q

;

%

INTEGER I

,J

,INDEX

,SNUM

,T2M

;

%

DEFINE

G(I,J) = A[B[I],J]*

,GETC = READ(CARD,/,FOR I:=1 STEP 1 UNTIL N DO C[I])#

,DOTS = BEGIN

```

GETC;
OUTTER(G(INDEX,*).C.SNUM.N);
END#

```

```

? .....
INTEGER PROCEDURE NUMB;
  BEGIN
    SCAN P:P UNTIL = "(";
    SCAN P:P+1 UNTIL IN ALPHA;
    SCAN Q:Q WHILE IN ALPHA;
    NUMB:=INTEGER(P.DELTA(P.Q));
  END NUMB;
? .....
PROCEDURE OUTTER(A,R,I,J);
  VALUE          I,J ;
  INTEGER         I,J ;
  ARRAY          A,R[0] ;
  BEGIN
    INTEGER K;
    CASE I OF
      BEGIN
        FOR K:=1 STEP 1 UNTIL J DO A[K]:=**R[K];
        FOR K:=1 STEP 1 UNTIL J DO A[K]:=**R[K];
        FOR K:=1 STEP 1 UNTIL J DO A[K]:=**R[K];
        FOR K:=1 STEP 1 UNTIL J DO A[K]:=**R[K];
        END CASE STATEMENT;
      END OUTTER;
? .....
INTEGER PROCEDURE SCANNER;
  BEGIN
    LABEL ENFF;
    READ(CARD,14.0[*1],(ENFF::));
    ENFF: SCANNER:= SNUM:= 7;
    SCAN P:P UNTIL = " ";
    SCANNER:=SNUM:=
      IF P = "ADD" THEN 0 ELSE
      IF P = "SUB" THEN 1 ELSE
      IF P = "MULT" THEN 2 ELSE
      IF P = "DIV" THEN 3 ELSE
      IF P = "OVER" THEN 4 ELSE
      IF P = "DEL" THEN 5 ELSE
      IF P = "INS" THEN 6 ELSE
      IF P = "SCAL" THEN 7 ELSE
      IF P = "ADJUN" THEN 8 ELSE 9;
    IF SNUM NEQ 7 AND SNUM NEQ 8 AND SNUM NEQ 9 THEN
      INDEX:=NUMB;
    END SCANNER;
? .....
?
? BEGIN EXECUTABLE EDITFILE CODE
?
? .....
?
? INITIALIZE
?
? T2M:=M+M;
? FOR I := 1 STEP 1 UNTIL M DO

```

```

      READ(FIL,/,FOR J := 1 STEP 1 UNTIL N DO A[I,J])(FOFA:);
FOFA:
  FOR I:= 1 STEP 1 UNTIL M DO B[I]:=I;
%
% MAIN PROGRAM
%
  WHILE TRUE DO CASE SCANNER OF
    BEGIN
      DOTS;
      DOTS;
      DOTS;
      DOTS;
      READ(CARD,F9,FOR I:=1 STEP 1 UNTIL N DO G(INDEX,I));
      BEGIN
        REPLACE POINTER(R[INDEX])BY POINTER(B[INDEX+1]) FOR
          T2M=INDEX-1 WORDS;
        M:=M-1;
      END;
      BEGIN
        B[INDEX]:=M:=M+1;
        FOR I:= M STEP -1 UNTIL INDEX DO R[I]:=B[I-1];
        READ(CARD,F9,FOR J:=1 STEP 1 UNTIL N DO A[M,J]);
      END;
    BEGIN
      READ(CARD,/,FOR I:= 1 STEP 1 UNTIL M DO E[I]);
      FOR I:= 1 STEP 1 UNTIL N DO
        FOR J:= 1 STEP 1 UNTIL M DO A[I,J]:= **E[J];
      END;
      BEGIN
        READ(CARD,/,FOR I:= 1 STEP 1 UNTIL M DO E[I]);
        FOR I:= 1 STEP 1 UNTIL N DO
          FOR J:= 1 STEP 1 UNTIL M DO A[I,J]:=**E[I];
        END;
      BEGIN
        GO OUT;
        END CASE;
      %
      OUT;
      %
      % WRAP-UP
      %
      REWIND(FIL);
      FOR I:=1 STEP 1 UNTIL M DO
        WRITE(FIL,/,FOR J:=1 STEP 1 UNTIL N DO G(I,J));
    END EDITFILE;
% =====
PROCEDURE ERGWORKS(I,J,K,L,M);
VALUE
  I,J,K,L,M ;
INTEGER
  I,J,K,L,M ;
BEGIN
  REAL ARRAY A[0:13], MU[0:L], Q[0:I], Y[0:J], R[0:L], SH[0:M],
    P[0:I,0:J], M[0:K,0:K], B[0:L,0:M], SB[0:K,0:M];
  INTEGER I, J, K, L, BRANCH, YEAR, COLUMN;
  BOOLEAN FULLPRINT;
  REAL SUM, TOTAL;
  LABEL FINISH;
% INTERPOLATION PROCEDURE FOR YEARS OTHER THAN '72 OR '76

```



```

X
PROCEDURE INTERPOLATE:
BEGIN
  REAL F;
  F := (YEAR - 72)/4;
  FOR I := 1 STEP 1 UNTIL II DO BEGIN
    READ(P1,/,FOR J := 1 STEP 1 UNTIL JJ DO P[I,J]);
    READ(P2,/,FOR J := 1 STEP 1 UNTIL JJ DO Y[I]);
    FOR J := 1 STEP 1 UNTIL JJ DO P[I,J] := P[I,J] + F * (Y[J] - P[I,J]);
  END;
  FOR I := 1 STEP 1 UNTIL KK DO BEGIN
    READ(M3,/,FOR J := 1 STEP 1 UNTIL KK DO M[J,I]);
    READ(M4,/,FOR J := 1 STEP 1 UNTIL KK DO Y[J]);
    FOR J := 1 STEP 1 UNTIL KK DO M[J,I] := M[J,I] + F * (Y[J] - M[J,I]);
  END;
  FOR I := 1 STEP 1 UNTIL LL DO BEGIN
    READ(R1,/,FOR J := 1 STEP 1 UNTIL MM DO R[I,J]);
    READ(R2,/,FOR J := 1 STEP 1 UNTIL MM DO SH[J]);
    FOR J := 1 STEP 1 UNTIL MM DO R[I,J] := R[I,J] + F * (SH[J] - R[I,J]);
  END;
  READ(MU1,/,FOR I := 1 STEP 1 UNTIL LL DO MU[I]);
  READ(MU2,/,FOR I := 1 STEP 1 UNTIL LL DO R[I]);
  FOR I := 1 STEP 1 UNTIL LL DO MU[I] := MU[I] + F * (R[I] - MU[I]);
END;
X
X      DEFINE SPECIAL LISTS FREQUENTLY USED FOR OUTPUT
X
X
DEFINE SPECIALROWS = J := 2, 3, 5, 6, 7, 8, 9, 11, 12, 16, 19, 20, 21,
24, 25, 28, 29, 30, 31, 34, 37, 38, 42, 43, 47, 48, 50, 51, 54,
55, 56, 58, 59, 60, 61, 62, 63, 65, LL#;
X
X
DEFINE SPECIALCOLUMNS = J := 2, 12, 20, 26, 38, 43, 47, 61, 69, 70, 71,
81, 83, 94, 95, 106, 112, 116, 123, 136, 139, 143, 153, 158, 167,
168, 172, 177, 184, MM#;
X
X      BEGIN PROCESSING BY READING IN OPTION CARD AND DATA
X
X
READ(CARD,/,YEAR,I,BRANCH);
IF I = 0 THEN FULLPRINT := FALSE ELSE FULLPRINT := TRUE;
IF YEAR GTR 1900 THEN YEAR := YEAR - 1900;
IF YEAR = 72 THEN BEGIN SEG;
  FOR I := 1 STEP 1 UNTIL II DO
    READ(P1,/,FOR J := 1 STEP 1 UNTIL JJ DO P[I,J]);
  FOR I := 1 STEP 1 UNTIL KK DO
    READ(M3,/,FOR J := 1 STEP 1 UNTIL KK DO M[J,I]);
  FOR I := 1 STEP 1 UNTIL LL DO
    READ(R1,/,FOR J := 1 STEP 1 UNTIL MM DO R[I,J]);
  READ(MU1,/,FOR I := 1 STEP 1 UNTIL LL DO MU[I]);
  END
  ELSE IF YEAR = 76 THEN BEGIN SEG;
  FOR I := 1 STEP 1 UNTIL II DO
    READ(P2,/,FOR J := 1 STEP 1 UNTIL JJ DO P[I,J]);
  FOR I := 1 STEP 1 UNTIL KK DO
    READ(M4,/,FOR J := 1 STEP 1 UNTIL KK DO M[J,I]);
  FOR I := 1 STEP 1 UNTIL LL DO
    READ(R2,/,FOR J := 1 STEP 1 UNTIL MM DO R[I,J]);
  READ(MU2,/,FOR I := 1 STEP 1 UNTIL LL DO MU[I]);

```


END

ELSE INTERPOLATE;

IF BRANCH GTR 3 OR BRANCH LSS 1 THEN BEGIN

WRITE(LINE,ERRMESS1);

GO TO FINISH;

END;

IF BRANCH = 3 THEN READ(CARD,/,COLUMN);

IF BRANCH = 3 AND (COLUMN LSS 1 OR COLUMN GTR MM) THEN BEGIN

WRITE(LINE,ERRMESS1);

GO TO FINISH;

END;

%
% READ AND WRITE TITLE AND Q-VECTOR
%

READ(CARD,13,A[*]);

WRITE(LINE,13,A[*]);

I := 0;

THRU ((II DIV 8))

DO READ(CARD,F8, FOR J := 1 STEP 1 UNTIL 8 DO Q[I:=I+1]);

READ(CARD,F8,THRU II MOD 8 DO Q[I:=I+1]);

WRITE(LINE,</" ALTERNATIVE EXPENDITURE VECTOR">);

WRITE(LINE,HEAD1);

TOTAL := 0;

FOR I := 1 STEP 1 UNTIL II DO BEGIN

TOTAL := TOTAL + Q[I];

WRITE(LINE,F1,I,Q[I]); END;

WRITE(LINE[SKIP 1],F2,TOTAL);

%
% FORM INDUSTRY ACTIVITY MATRIX (OVERLAYING P)
%

FOR I := 1 STEP 1 UNTIL II DO

FOR J := 1 STEP 1 UNTIL JJ DO P[I,J] := P[I,J] + Q[I];

%
% ROUTINE TO PRINT INDUSTRY ACTIVITY MATRIX AND ITS TRANSPOSE
%

IF FULLPRINT THEN BEGIN

WRITE(LINE,<"INDUSTRY ACTIVITY MATRIX">);

FOR K := 1 STEP 10 UNTIL 41 DO BEGIN

L := K + 9;

WRITE(LINE,HEAD4,K,L);

FOR J := 1 STEP 1 UNTIL JJ DO

WRITE(LINE,F10,J, FOR I := K STEP 1 UNTIL L DO P[I,J]);

END;

WRITE(LINE,HEAD4,51,II);

FOR J := 1 STEP 1 UNTIL JJ DO

WRITE(LINE,F7,J, FOR I := 51 STEP 1 UNTIL II DO P[I,J]);

WRITE(LINE[SKIP 1]);

WRITE(LINE,<"TRANSPOSE PRINT OF INDUSTRY ACTIVITY MATRIX">);

FOR K := 1 STEP 10 UNTIL 71 DO BEGIN

L := K + 9;

WRITE(LINE,HEAD5,K,L);

FOR J := 1 STEP 1 UNTIL II DO

WRITE(LINE,F10,J, FOR I := K STEP 1 UNTIL L DO P[J,I]);

END;

WRITE(LINE,HEAD5,81,JJ);

FOR J := 1 STEP 1 UNTIL II DO

WRITE(LINE,F10,J, FOR I := 81 STEP 1 UNTIL JJ DO P[J,I]);

```

WRITE(LINE[SKIP 1]);
      END;
%
%      GENERATE FINAL DEMAND VECTOR AND PRINT
%
WRITE(LINE,<" GENERATED FINAL DEMAND VECTOR">); WRITE(LINE,HEAD1);
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL JJ DO BEGIN
  Y[I] := 0;
  FOR J := 1 STEP 1 UNTIL II DO Y[I] := Y[I] + P[J,I];
  WRITE(LINE,F1,I,Y[I]);
  TOTAL := TOTAL + Y[I];      END;
WRITE(LINE[SKIP 1],F2,TOTAL);
%
%      AGGREGATE FINAL DEMAND VECTOR AND PRINT
%
WRITE(LINE,<" AGGREGATE FINAL DEMAND VECTOR">); WRITE(LINE,HEAD1);
Y[R0] := Y[R1];
Y[R1] := Y[R2];
Y[R2] := Y[R7];
Y[R3] := Y[R4];
Y[R4] := Y[R6];
Y[KK] := Y[J,J];
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL KK DO BEGIN
  TOTAL := TOTAL + Y[I];
  WRITE(LINE,F1,I,Y[I]);      END;
WRITE(LINE[SKIP 1],F2,TOTAL);
%
%      GENERATE THE INTERINDUSTRY EMPLOYMENT MATRIX
%
WRITE(LINE,<" ROW AND COLUMN SUMS OF INTERINDUSTRY EMPLOYMENT MATRIX">);
WRITE(LINE,HEAD3);
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL KK DO BEGIN
  SUM := 0;
  FOR J := 1 STEP 1 UNTIL KK DO BEGIN
    M[I,J] := M[I,J] * Y[I];
    SUM := SUM + M[I,J];
    END;
  WRITE(LINE,F1,I,SUM);
  TOTAL := TOTAL + SUM;      END;
WRITE(LINE[SPACE 2],F2,TOTAL);
WRITE(LINE,HEAD2);
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL KK DO BEGIN
  SUM := 0;
  FOR J := 1 STEP 1 UNTIL KK DO SUM := SUM + M[J,I];
  WRITE(LINE,F1,I,SUM);
  TOTAL := TOTAL + SUM;      END;
WRITE(LINE[SKIP 1],F2,TOTAL);
%
%      ROUTINE TO PRINT INTERINDUSTRY EMPLOYMENT MATRIX AND TRANSPOSE
%
IF FULLPRINT THEN BEGIN
WRITE(LINE,<" INTERINDUSTRY EMPLOYMENT MATRIX">);
FOR K := 1 STEP 10 UNTIL 71 DO BEGIN

```

```

L := K + 9;
WRITE(LINE, HEAD4, K, L);
FOR J := 1 STEP 1 UNTIL KK DO
    WRITE(LINE, F10, J, FOR I := K STEP 1 UNTIL L DO M[I, J]);
    END;
WRITE(LINE, HEAD4, 81, KK);
FOR J := 1 STEP 1 UNTIL KK DO
    WRITE(LINE, F7, J, FOR I := 81 STEP 1 UNTIL KK DO M[I, J]);
WRITE(LINE[SKIP 1]);
%
WRITE(LINE, <" TRANSPOSE PRINT OF INTERINDUSTRY EMPLOYMENT MATRIX">);
FOR K := 1 STEP 10 UNTIL 71 DO BEGIN
    L := K + 9;
    WRITE(LINE, HEAD5, K, L);
    FOR J := 1 STEP 1 UNTIL KK DO
        WRITE(LINE, F10, J, FOR I := K STEP 1 UNTIL L DO M[J, I]);
        END;
    WRITE(LINE, HEAD5, 81, KK);
    FOR J := 1 STEP 1 UNTIL KK DO
        WRITE(LINE, F7, J, FOR I := 81 STEP 1 UNTIL KK DO M[J, I]);
    WRITE(LINE[SKIP 1]);
    END;
%
%      AGGREGATE THE INTERINDUSTRY EMPLOYMENT MATRIX TO LL X KK
%      (OVERLAYING ITSELF) AND TAKE THE ROW AND COLUMN SUMS
%
FOR I := 1 STEP 1 UNTIL KK DO BEGIN SEG;
    M[I, 2] := M[I, 1] + M[I, 2];
    M[I, 3] := M[I, 3] + M[I, 4];
    M[I, 1] := M[I, 2] + M[I, 3];
    M[I, 5] := M[I, 5] + M[I, 6];
    M[I, 6] := M[I, 7];
    M[I, 7] := M[I, 8];
    M[I, 8] := M[I, 9] + M[I, 10];
    M[I, 4] := M[I, 5] + M[I, 6] + M[I, 7] + M[I, 8];
    M[I, 9] := M[I, 11] + M[I, 12] + M[I, 85];
    M[I, 10] := SUM := M[I, 13];
    FOR J := 14 STEP 1 UNTIL 64 DO M[I, 10] := M[I, 10] + M[I, J];
    M[I, 11] := M[I, 14];
    M[I, 12] := M[I, 15];
    M[I, 13] := M[I, 16] + M[I, 17];
    M[I, 14] := M[I, 16];
    M[I, 15] := M[I, 17];
    M[I, 16] := M[I, 18] + M[I, 19];
    M[I, 17] := M[I, 18];
    M[I, 18] := M[I, 19];
    M[I, 19] := M[I, 20] + M[I, 21];
    M[I, 20] := M[I, 22] + M[I, 23];
    M[I, 21] := M[I, 24] + M[I, 25];
    M[I, 22] := M[I, 25];
    M[I, 23] := M[I, 24];
    M[I, 24] := M[I, 26];
    M[I, 26] := M[I, 27] + M[I, 28] + M[I, 29];
    M[I, 27] := M[I, 30];
    M[I, 25] := M[I, 26] + M[I, 27];
    M[I, 28] := M[I, 31];
    M[I, 29] := M[I, 32];

```

```

M[I,30] := M[I,33] + M[I,34];
M[I,31] := M[I,35] + M[I,36];
M[I,32] := M[I,35];
M[I,33] := M[I,36];
M[I,34] := M[I,37] + M[I,38];
M[I,35] := M[I,37];
M[I,36] := M[I,38];
M[I,37] := SUM + M[I,39] + M[I,40] + M[I,41] + M[I,42];
M[I,38] := M[I,43];
FOR J := 44 STEP 1 UNTIL 52 DO M[I,38] := M[I,38] + M[I,J];
M[I,39] := M[I,44];
M[I,40] := M[I,51];
M[I,41] := M[I,38] - M[I,39] - M[I,40];
M[I,42] := M[I,53] + M[I,54] + M[I,55] + M[I,56] + M[I,57] + M[I,58];
M[I,43] := M[I,59] + M[I,60] + M[I,61];
M[I,44] := M[I,59];
M[I,45] := M[I,60];
M[I,46] := M[I,61];
M[I,47] := M[I,62] + M[I,63];
M[I,48] := M[I,64];
M[I,50] := M[I,65];
M[I,51] := M[I,66] + M[I,67];
M[I,52] := M[I,67];
M[I,53] := M[I,66];
M[I,54] := M[I,68];
M[I,49] := M[I,50] + M[I,51] + M[I,54];
M[I,55] := M[I,69];
M[I,56] := M[I,70] + M[I,71];
M[I,57] := M[I,72] + M[I,73] + M[I,74] + M[I,75] + M[I,76] + M[I,77];
M[I,58] := M[I,72];
M[I,59] := M[I,73] + M[I,74];
M[I,60] := M[I,75];
M[I,61] := M[I,76];
M[I,62] := M[I,77];
M[I,63] := M[I,84];
M[I,65] := M[I,78] + M[I,82];
M[I,66] := M[I,79] + M[I,83];
M[I,64] := M[I,65] + M[I,66];

```

END;

```

%
% ROUTINE TO PRINT OUT AGGREGATED INTERINDUSTRY
% EMPLOYMENT MATRIX AND ROW AND COLUMN SUMS
%
WRITE(LINE, <" AGGREGATED INTERINDUSTRY EMPLOYMENT MATRIX">);
WRITE(LINE, <" (ROW AND COLUMN SUMS OVER SELECTED ROWS)">);
WRITE(LINE, HEAD2);
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL LL DO BEGIN
  R[I] := 0;
  FOR J := 1 STEP 1 UNTIL KK DO R[I] := R[I] + M[J, I];
  WRITE(LINE, F1, I, R[I]);
  END;
FOR SPECIALROWS DO TOTAL := TOTAL + R[J];
WRITE(LINE, SPACE 2, F2, TOTAL);
WRITE(LINE, HEAD3);
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL KK DO BEGIN
  Y[I] := 0;

```

```

FOR SPECIALROWS DO Y[I] := Y[I] + M[I,J];
WRITE(LINE,F1,I,Y[I]);
TOTAL := TOTAL + Y[I];      END;
WRITE(LINE[SKIP 1],F2,TOTAL);
%
%      ROUTINE TO PRINT AGGREGATE INTERINDUSTRY EMPLOYMENT MATRIX
%
IF FULLPRINT THEN BEGIN
WRITE(LINE,<"AGGREGATE INTERINDUSTRY EMPLOYMENT MATRIX">);
FOR K := 1 STEP 10 UNTIL 71 DO BEGIN
  L := K + 9;
  WRITE(LINE,HEAD4,K,L);
  FOR J := 1 STEP 1 UNTIL LL DO
    WRITE(LINE,F10,J,FOR I := K STEP 1 UNTIL L DO M[I,J]);
  END;
WRITE(LINE,HEAD4,81,KK);
FOR J := 1 STEP 1 UNTIL LL DO
  WRITE(LINE,F7,J,FOR I := 81 STEP 1 UNTIL KK DO M[I,J]);
WRITE(LINE[SKIP 1]);
%
WRITE(LINE,<"AGGREGATE INTERINDUSTRY EMPLOYMENT TRANSPOSE">);
FOR K := 1 STEP 10 UNTIL 51 DO BEGIN
  L := K + 9;
  WRITE(LINE,HEAD5,K,L);
  FOR J := 1 STEP 1 UNTIL KK DO
    WRITE(LINE,F10,J,FOR I := K STEP 1 UNTIL L DO M[J,I]);
  END;
WRITE(LINE,HEAD5,61,LL);
FOR J := 1 STEP 1 UNTIL KK DO
  WRITE(LINE,F7,J,FOR I := 61 STEP 1 UNTIL LL DO M[J,I]);
WRITE(LINE[SKIP 1]);
  ENN;
%
%      FOR BRANCHES 2 AND 3 MODIFY THE M-MATRIX WITH MU
%
IF BRANCH GTR 1 THEN BEGIN
  WRITE(LINE,HEAD2);
  TOTAL := 0;
  FOR I := 1 STEP 1 UNTIL LL DO BEGIN
    SUM := 0;
    FOR J := 1 STEP 1 UNTIL KK DO BEGIN
      M[J,I] := M[J,I] + MU[I];
      SUM := SUM + M[J,I];      ENN;
    END;
    WRITE(LINE,F1,I,SUM);
    TOTAL := TOTAL + SUM;      END;
  WRITE(LINE[SKIP 1],F2,TOTAL);
%
%      ROUTINE TO PRINT MODIFIED M-MATRIX
%
IF FULLPRINT THEN BEGIN
  WRITE(LINE,<"MODIFIED INTERINDUSTRY EMPLOYMENT MATRIX">);
  FOR K := 1 STEP 10 UNTIL 71 DO BEGIN
    L := K + 9;
    WRITE(LINE,HEAD4,K,L);
    FOR J := 1 STEP 1 UNTIL LL DO
      WRITE(LINE,F10,J,FOR I := K STEP 1 UNTIL L DO M[I,J]);
    END;

```

```

WRITE(LINE,HEAD4,81,KK);
FOR J := 1 STEP 1 UNTIL LL DO
  WRITE(LINE,F7,J,FOR I := 81 STEP 1 UNTIL KK DO M[I,J]);
WRITE(LINE[SKIP 1]);
WRITE(LINE,<"TRANSPONSE PRINT OF MODIFIED M-MATRIX">);
FOR K := 1 STEP 10 UNTIL 51 DO BEGIN
  L := K + 9;
  WRITE(LINE,HEAD5,K,L);
  FOR J := 1 STEP 1 UNTIL KK DO
    WRITE(LINE,F10,J,FOR I := K STEP 1 UNTIL L DO M[J,I]);
  END;
WRITE(LINE,HEAD5,61,LL);
FOR J := 1 STEP 1 UNTIL KK DO
  WRITE(LINE,F7,J,FOR I := 61 STEP 1 UNTIL LL DO M[J,I]);
WRITE(LINE[SKIP 1]);
END;
END;

%
%           SELECT BRANCH FOR REMAINING PROCESSING
%
CASE BRANCH OF BEGIN;
%
%           B R A N C H       U N E
%           - - - - - - - - -
%
%   BEGIN BY USING THE MU VECTOR TO MODIFY R
%
BEGIN SEG;
WRITE(LINE[SPACE 2],<"***NEW R-VECTOR (MODIFIED BY MU)">);
WRITE(LINE,HEAD1);
FOR I := 1 STEP 1 UNTIL LL DO BEGIN
  R[I] := R[I] * MU[I];
  WRITE(LINE,F1,I,R[I]);
END;
TOTAL := 0;
FOR SPECIALROWS DO TOTAL := TOTAL + R[J];
WRITE(LINE[SKIP 1],F2,TOTAL);
%
%   FORM S(ALPHA), OVERLAYING THE B-MATRIX
FOR I := 1 STEP 1 UNTIL LL DO
  FOR J := 1 STEP 1 UNTIL MM DO B[I,J] := R[I] * B[I,J];
%
%   ROUTINE TO PRINT S(ALPHA) AND ITS TRANSPONSE
%
IF FULLPRINT THEN BEGIN
WRITE(LINE,<"***S(ALPHA) MATRIX">);
FOR K := 1 STEP 10 UNTIL 171 DO BEGIN
  L := K + 9;
  WRITE(LINE,HEAD4,K,L);
  FOR J := 1 STEP 1 UNTIL LL DO
    WRITE(LINE,F10,J,FOR I := K STEP 1 UNTIL L DO B[J,I]);
  END;
WRITE(LINE,HEAD4,181,MM);
FOR J := 1 STEP 1 UNTIL LL DO
  WRITE(LINE,F7,J,FOR I := 181 STEP 1 UNTIL MM DO B[J,I]);
WRITE(LINE[SKIP 1]);
%

```



```

WRITE(LINE,<"TRANPOSE PRINT OF S(ALPHA)">);
FOR K := 1 STEP 10 UNTIL 51 DO BEGIN
  L := K + 9;
  WRITE(LINE,HEAD5,K,L);
  FOR J := 1 STEP 1 UNTIL MM DO
    WRITE(LINE,F10,J,FOR I := K STEP 1 UNTIL L DO B[I,J]);
  END;
WRITE(LINE,HEAD5,61,LL);
FOR J := 1 STEP 1 UNTIL MM DO
  WRITE(LINE,F7,J,FOR I := 61 STEP 1 UNTIL LL DO B[I,J]);
WRITE(LINE[SKIP 1]);
END;

%
%   CALCULATE ROWSUMS OVER SPECIAL COLUMNS, OVERLAYING R
%
WRITE(LINE,<"GENERATED INDUSTRY EMPLOYMENT VECTOR">);
WRITE(LINE,HEAD1);
FOR I := 1 STEP 1 UNTIL LL DO BEGIN
  R[I] := 0;
  FOR SPECIALCOLUMNS DO R[I] := R[I] + B[I,J];
  WRITE(LINE,F1,I,R[I]);
  END;
TOTAL := 0;
FOR SPECIALROWS DO TOTAL := TOTAL + R[J];
WRITE(LINE[SPACE 2],</"TOTAL EMPLOYMENT = ",F17.5>,TOTAL);
WRITE(LINE,<"GENERATED OCCUPATIONAL EMPLOYMENT VECTOR">);
WRITE(LINE,HEAD1);
%
%   TAKE COLUMNSUMS OVER SELECTED ROWS AND PRINT
%
FOR I := 1 STEP 1 UNTIL MM DO BEGIN
  SH[I] := 0;
  FOR SPECIALROWS DO SH[I] := SH[I] + B[J,I];
  WRITE(LINE,F1,I,SH[I]);
  END;
TOTAL := 0;
FOR SPECIALCOLUMNS DO TOTAL := TOTAL + SH[J];
WRITE(LINE,</"TOTAL EMPLOYMENT = ",F17.5>,TOTAL);
END OF BRANCH ONE;
%
%
%   B R A N C H   T W O
%   - - - - -
%
%
%   MULTIPLY M * B TO GET S(BETA), OVERLAYING B.
%   HIT ONLY SPECIAL ROWS TO AVOID DOUBLE-COUNTING.
%
BEGIN SEG;
FOR I := 1 STEP 1 UNTIL KK DO FOR K := 1 STEP 1 UNTIL MM DO BEGIN
  SB[I,K] := 0;
  FOR SPECIALROWS DO SB[I,K] := SB[I,K] + M[I,J] * B[J,K];
  END;
%   ROUTINE TO PRINT S(BETA), AND TRANSPOSE
%
IF FULLPRINT THEN BEGIN
WRITE(LINE,<"***S(BETA) MATRIX">);
FOR K := 1 STEP 10 UNTIL 171 DO BEGIN
  L := K + 9;

```

```

WRITE(LINE,HEAD4,K,L);
FOR J := 1 STEP 1 UNTIL KK DO
  WRITE(LINE,F10,J,FOR I := K STEP 1 UNTIL L DO SB(J,I));
END;

WRITE(LINE,HEAD4,181,MM);
FOR J := 1 STEP 1 UNTIL KK DO
  WRITE(LINE,F7,J,FOR I := 181 STEP 1 UNTIL MM DO SB(J,I));
WRITE(LINE(SKIP 1));

WRITE(LINE,<"TRANSPOSE PRINT OF S(BETA)">);
FOR K := 1 STEP 10 UNTIL 71 DO BEGIN
  L := K + 9;
  WRITE(LINE,HEAD5,K,L);
  FOR J := 1 STEP 1 UNTIL MM DO
    WRITE(LINE,F10,J,FOR I := K STEP 1 UNTIL L DO SB(I,J));
  END;

  WRITE(LINE,HEAD5,81,KK);
  FOR J := 1 STEP 1 UNTIL MM DO
    WRITE(LINE,F7,J,FOR I := 81 STEP 1 UNTIL KK DO SB(I,J));
  WRITE(LINE(SKIP 1));
  END;

  CALCULATE VECTOR OF COLUMNSUMS AND PRINT

  WRITE(LINE,<"TOTAL OCCUPATIONAL EMPLOYMENT GENERATED BY:"/>);
  FOR I := 1 STEP 1 UNTIL MM DO BEGIN
    SH[I] := SB[1,I];
    FOR J := 2 STEP 1 UNTIL KK DO SH[I] := SH[I] + SB[J,I];
    WRITE(LINE,F1,I,SH[I]);
  END;
  TOTAL := 0;
  FOR SPECIALCOLUMNS DO TOTAL := TOTAL + SH[J];
  WRITE(LINE(SPACE 4),F2,TOTAL);

  CALCULATE AND PRINT VECTOR OF ROWSUMS (OVER SPECIAL COLUMNS)

  WRITE(LINE,<"TOTAL INDUSTRIAL EMPLOYMENT GENERATED BY:"/>);
  TOTAL := 0;
  FOR I := 1 STEP 1 UNTIL KK DO BEGIN
    SUM := 0;
    FOR SPECIALCOLUMNS DO SUM := SUM + SB[I,J];
    WRITE(LINE,F1,I,SUM);
    TOTAL := TOTAL + SUM;
  END;
  WRITE(LINE,F2,TOTAL);
END OF BRANCH TWO;

BRANCH THREE
- - - - -

PULL OUT A COLUMN VECTOR FROM B, OVERWRITING R, AND PRINT IT

BEGIN SEG;
WRITE(LINE,<"SELECTED COLUMN VECTOR FROM B-MATRIX">);
WRITE(LINE,HEAD1);
FOR I := 1 STEP 1 UNTIL LL DO BEGIN
  R[I] := B[I,COLUMN];
  WRITE(LINE,F1,I,R[I]);
END;

```



```
%
%      FORM S(H) MATRIX, OVERWRITING M
%
FOR I := 1 STEP 1 UNTIL KK DO FOR J := 1 STEP 1 UNTIL LL DO
  M[I,J] := M[I,J] + R[J];
%
%      ROUTINE TO PRINT S(H)
%
IF FULLPRINT THEN BEGIN
  WRITE(LINE[SKIP 1]);
  WRITE(LINE,<"***S(H) MATRIX**>");
  FOR K := 1 STEP 10 UNTIL 51 DO BEGIN
    L := K + 9;
    WRITE(LINE,HEAD4,K,L);
    FOR J := 1 STEP 1 UNTIL KK DO
      WRITE(LINE,F10,J,FOR I := K STEP 1 UNTIL L DO M[J,I]);
    END;
  WRITE(LINE,HEAD4,61,LL);
  FOR J := 1 STEP 1 UNTIL KK DO
    WRITE(LINE,F7,J,FOR I := 61 STEP 1 UNTIL LL DO M[J,I]);
  WRITE(LINE[SKIP 1]);
%
  WRITE(LINE,<"TRANSPOSE PRINT OF S(H)**>");
  FOR K := 1 STEP 10 UNTIL 71 DO BEGIN
    L := K + 9;
    WRITE(LINE,HEAD5,K,L);
    FOR J := 1 STEP 1 UNTIL LL DO
      WRITE(LINE,F10,J,FOR I := K STEP 1 UNTIL L DO M[I,J]);
    END;
  WRITE(LINE,HEAD5,81,KK);
  FOR J := 1 STEP 1 UNTIL LL DO
    WRITE(LINE,F7,J,FOR I := 81 STEP 1 UNTIL KK DO M[I,J]);
  WRITE(LINE[SKIP 1]);
    END;
%
%      COMPUTE COLUMNSUMS OF S(H), OVERWRITING R, AND PRINT
%
WRITE(LINE,<"EMPLOYMENT GENERATED IN:**>");
FOR I := 1 STEP 1 UNTIL LL DO BEGIN
  R[I] := M[1,I];
  FOR J := 2 STEP 1 UNTIL KK DO R[I] := R[I] + M[J,I];
  WRITE(LINE,F1,I,R[I]);
  END;
%
%      SUM THE COLUMNSUMS FOR SPECIAL COLUMNS
%
TOTAL := 0;
FOR SPECIALROWS DO TOTAL := TOTAL + R[J];
WRITE(LINE[SPACE 4],F2,TOTAL);
%
%      COMPUTE ROWSUMS OVER SPECIAL COLUMNS, PRINT, AND TOTAL
%
WRITE(LINE,<"EMPLOYMENT GENERATED BY:**>");
TOTAL := 0;
FOR I := 1 STEP 1 UNTIL KK DO BEGIN
  SUM := 0;
  FOR SPECIALROWS DO SUM := SUM + M[I,J];
  WRITE(LINE,F1,I,SUM);
```

```

TOTAL := TOTAL + SUM;          END;
WRITE(LINE,F2,TOTAL);
END OF BRANCH THREE;
    END;
FINISH: END ERGWORKS;
    L:=57;
    FOR I:= 1 STEP 1 UNTIL L DO
    READ(P1,/,FOR J:= 1 STEP 1 UNTIL 89 DO P[I,J]);
    REWIND(P1);
    WRITE(LINE,<"INDUSTRY ACTIVITY MATRIX">);
    PRTMATRIX(P,L,89,0);
    WRITE(LINE,<"TRANSPOSE OF INDUSTRY ACTIVITY MATRIX">);
    PRITRANS(P,L,89,0);
    EDITFILF(P1,L,89);
    REWIND(P1);
    FOR I:= 1 STEP 1 UNTIL L DO
    READ(P1,/,FOR J:=1 STEP 1 UNTIL 89 DO P[I,J]);
    REWIND(P1);
    WRITE(LINE,<"MODIFIED P MATRIX">);
    PRTMATRIX(P,L,89,0);
    WRITE(LINE,<"TRANSPOSE OF P MATRIX">);
    PRITRANS(P,L,89,0);
    EDITFILF(P1,L,89);
    REWIND(P1);
    FOR I:= 1 STEP 1 UNTIL L DO
    READ(P1,/,FOR J:= 1 STEP 1 UNTIL 89 DO P[I,J]);
    REWIND(P1);
    WRITE(LINE,<"SCALED P MATRIX">);
    PRTMATRIX(P,L,89,0);
    WRITE(LINE,<"TRANSPOSE OF SCALED P MATRIX">);
    PRITRANS(P,L,89,0);
    FOR I := 1 STEP 1 UNTIL 85 DO
    READ(M3,/,FOR J := 1 STEP 1 UNTIL 85 DO M[J,I]);
    REWIND(M3);
    WRITE(LINE,<"M-MATRIX">);
    PRTMATRIX(M,85,85,0);
    WRITE(LINE,<"TRANSPOSE OF M-MATRIX">);
    PRITRANS(M,85,85,0);
    EDITFILF(M3,85,85);
    REWIND(M3);
    FOR I:= 1 STEP 1 UNTIL 85 DO
    READ(M3,/,FOR J:= 1 STEP 1 UNTIL 85 DO M[J,I]);
    REWIND(M3);
    WRITE(LINE,<"MODIFIED M-MATRIX">);
    PRTMATRIX(M,85,85,0);
    WRITE(LINE,<"TRANSPOSE OF MODIFIED M-MATRIX">);
    PRITRANS(M,85,85,0);
    ERGWORKS(L,89,85,66,185);
END.

```

Appendix D

Sample Input and Output

TEST DATA

ALTERNATIVE EXPENDITURE VECTOR

ELEMENT	VALUE		
1	103351.00000		
2	48160.00000	53	6824.00000
3	8099.00000	54	687.00000
4	75406.00000	55	770.00000
5	72094.00000	56	1035.00000
6	21283.00000	57	4681.00000
7	24001.00000	58	12415.00000
8	67069.00000	TOTAL	774509.00000
9	20564.00000		
10	6402.00000		
11	7083.00000		
12	3587.00000		
13	62804.00000		
14	25794.00000		
15	27440.00000		
16	5906.00000		
17	-6500.00000		
18	4600.00000		
19	3906.00000		
20	797.00000		
21	3834.00000		
22	14176.00000		
23	19391.00000		
24	347.00000		
25	9299.00000		
26	313.00000		
27	2802.00000		
28	0.00000		
29	0.00000		
30	1374.00000		
31	370.00000		
32	290.00000		
33	41937.00000		
34	706.00000		
35	907.00000		
36	4281.00000		
37	173.00000		
38	304.00000		
39	667.00000		
40	1548.00000		
41	879.00000		
42	1002.00000		
43	2200.00000		
44	3408.00000		
45	2250.00000		
46	439.00000		
47	0.00000		
48	1474.00000		
49	5084.00000		
50	32480.00000		
51	9112.00000		
52	5092.00000		

ROW AND COLUMN SUMS OF INTERINDUSTRY EMPLOYMENT MATRIX

COLUMN	COLUMNSUM		
1	289249.77572	55	211199.49912
2	537353.97308	56	921626.72039
3	35981.55789	57	104564.96353
4	10166.31912	58	90010.48281
5	-1064.87117	59	2573763.80533
6	1912.45540	60	1045679.12905
7	30317.41706	61	527922.62511
8	-5886.12880	62	299466.22780
9	85048.71956	63	236772.88622
10	1289.42814	64	552708.87146
11	1965798.72260	65	1644774.58678
12	374313.86243	66	630635.89490
13	508285.09958	67	2731.02644
14	5320377.88470	68	1002116.42013
15	325388.38505	69	16244997.38864
16	105752.50661	70	2495356.68854
17	131263.56008	71	2445661.58203
18	2564094.45276	72	3192606.44809
19	277697.53723	73	1012776.64438
20	570368.05696	74	95040.52720
21	1091.82242	75	828031.42501
22	595306.97675	76	748191.33627
23	275403.17427	77	7640221.48395
24	202781.86626	78	242704.68015
25	16843.36638	79	96315.54364
26	494042.14617	80	35030.16712
27	161160.03176	81	34286.15020
28	11468.31253	82	2023994.30758
29	631403.32544	83	10096654.92532
30	30103.06482	84	2547412.40765
31	651283.80363	85	2078297.53629
32	288463.25600	TOTAL	84032529.14160
33	285.99643		
34	475570.09785	ROW	ROWSUM
35	50268.18864	1	1689254.97548
36	534894.53473	2	1839631.48200
37	172235.51539	3	119099.17653
38	108967.18843	4	235582.09940
39	7751.40235	5	27595.21607
40	852897.71098	6	58178.57805
41	78696.04070	7	123171.32820
42	237967.47212	8	286566.31612
43	118575.01038	9	115260.33629
44	175247.30404	10	13001.09907
45	279396.60373	11	1050342.33251
46	168259.04752	12	1489505.79829
47	287759.33740	13	236054.50093
48	306600.47697	14	1082099.26771
49	252713.35514	15	76622.19367
50	21790.75311	16	564217.45394
51	527287.10543	17	117455.35275
52	289403.61609	18	1586577.00189
53	419723.14205	19	178935.59661
54	477628.05799	20	642782.66110
		21	33679.64879
		22	370987.58207

23	161007.35991	80	0.00000
24	487571.32193	81	0.00000
25	225252.80601	82	2023994.30758
26	1164931.63738	83	10096654.92532
27	474020.68407	84	2547412.40765
28	226722.49708	85	2078297.53629
29	276272.93052	TOTAL	84032529.14087
30	69748.11789		
31	167575.62322		
32	590948.60400		
33	29445.70660		
34	315756.83767		
35	191531.80310		
36	522064.21735		
37	895524.87708		
38	419996.11301		
39	78151.06769		
40	528035.64371		
41	358118.80132		
42	471928.02132		
43	111551.96437		
44	153200.04409		
45	195454.59357		
46	98248.77906		
47	359779.30353		
48	217225.64407		
49	290724.30547		
50	251521.27991		
51	267002.43727		
52	140336.42075		
53	420252.65377		
54	183522.05152		
55	209487.42303		
56	605590.87959		
57	384658.95282		
58	113070.49636		
59	876103.57793		
60	693153.37233		
61	312482.61370		
62	314433.66011		
63	161033.84430		
64	471559.05764		
65	2081556.23306		
66	904866.54076		
67	128104.71373		
68	672700.36969		
69	17098258.37500		
70	3045807.55504		
71	871355.66030		
72	3117411.53192		
73	3054421.97745		
74	100170.44155		
75	567692.22401		
76	823902.62573		
77	6256083.48122		
78	744255.54063		
79	530991.72139		

GENERATED INDUSTRY EMPLOYMENT VECTOR

ELEMENT

VALUE

1 4253482.02408
2 4050698.76459
3 212684.44737
4 656771.45303
5 86768.77113
6 125179.12185
7 299547.77124
8 145294.55393
9 5354278.18346
10 19245413.50668
11 1139454.55655
12 83679.10771
13 650556.69326
14 546218.91718
15 104406.06306
16 1739559.63271
17 1568643.72133
18 170937.17544
19 630462.87201
20 506991.65621
21 749819.60293
22 247232.53131
23 512539.76146
24 1313926.36362
25 1061732.96614
26 951998.91107
27 79715.12394
28 169569.89397
29 546922.93299
30 356180.07363
31 663577.58910
32 175431.42817
33 488132.51129
34 1319025.43111
35 234569.76173
36 375091.12106
37 1747373.81349
38 1298576.23357
39 151193.12431
40 249106.47301
41 1598476.10675
42 1219350.18641
43 1839112.54886
44 789116.49263
45 605122.32415
46 414351.15749
47 481465.00611
48 461556.10771
49 5213345.13325
50 3024533.22465
51 970165.64424
52 111631.23511
53 859594.71549
54 1149511.24619

55 17626594.55434
56 3756954.90832
57 19712795.15600
58 3081561.20932
59 2781414.13509
60 544700.68494
61 768948.28593
62 12508761.45374
63 1769177.91711
64 4801089.30404
65 2245327.45514
66 2509187.37315

TOTAL EMPLOYMENT = 61089317.42

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Center for Advanced Computation University of Illinois at Urbana-Champaign Urbana, Illinois 61801		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	
		2b. GROUP	
3. REPORT TITLE ECONOMIC RESEARCH GROUP WORKING PAPER NO. 5 The CAC Economic and Manpower Forecasting Model: Documentation and User's Guide			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Research Report			
5. AUTHOR(S) (First name, middle initial, last name) Roger H. Bezdek, R. Michael Lefler, Albert L. Meyers, Janet H. Spoonamore			
6. REPORT DATE October 15, 1971	7a. TOTAL NO. OF PAGES 63	7b. NO. OF REFS 9	
8a. CONTRACT OR GRANT NO. DAHCO4 72-C-0001	8a. ORIGINATOR'S REPORT NUMBER(S) CAC Document No. 15		
b. PROJECT NO. ARPA Order 1899			
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)		
d.			
10. DISTRIBUTION STATEMENT Copies may be obtained from the address given in (1) above. Approved for public release; distribution unlimited.			
11. SUPPLEMENTARY NOTES None		12. SPONSORING MILITARY ACTIVITY U.S. Army Research Office-Durham Duke Station Durham, North Carolina	
13. ABSTRACT This paper presents the preliminary documentation and user's guide for the Center for Advanced Computation economic and manpower forecasting model. Section I gives introductory and background information on the development of the model and presents a brief but rigorous theoretical basis for the on-line system. Section II gives a description of the basic MANPOWER/DEMAND program indicating the function of the program, the detailed workings of the system options, and the language in which it is written. Appendices contain specifications of the data tapes and disc files involved, flow charts of the computer processes, and sample data input and output.			

DD FORM 1473
1 NOV 66

UNCLASSIFIED

Security Classification

UNCLASSIFIED

Security Classification

14	KEY WORDS	LINK A		LINK B		LINK C	
		ROLE	WT	ROLE	WT	ROLE	WT
	Applications Social and Behavioral Sciences Economics						

UNCLASSIFIED

Security Classification



UNIVERSITY OF ILLINOIS-URBANA

510 841L63C C001
CAC DOCUMENTS\$URBANA
11-20 1971



3 0112 007263806